

ESTRATEGIAS PARA LA CONFORMACIÓN DEL GOBIERNO Y LA
INTEGRACIÓN DE COMPONENTES EN LA CONSTRUCCIÓN DE UN SISTEMA
ORIENTADO A SERVICIOS, DEFINIDAS A TRAVÉS DE UN CONJUNTO DE
BUENAS PRÁCTICAS PARA ARQUITECTOS DE SOFTWARE

HUGO FABIAN SANCHEZ BENITEZ

UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BOGOTÁ D.C.
2015

ESTRATEGIAS PARA LA CONFORMACIÓN DEL GOBIERNO Y LA
INTEGRACIÓN DE COMPONENTES EN LA CONSTRUCCIÓN DE UN SISTEMA
ORIENTADO A SERVICIOS, DEFINIDAS A TRAVÉS DE UN CONJUNTO DE
BUENAS PRÁCTICAS PARA ARQUITECTOS DE SOFTWARE

HUGO FABIAN SANCHEZ BENITEZ
CÓDIGO 1110072

ASESORES
GILBERTO PEDRAZA GARCÍA
INGENIERO DE SISTEMAS Y COMPUTACIÓN

IGNACIO HERNÁNDEZ MOLINA
SOCIÓLOGO

UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BOGOTÁ D.C.
2015

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Bogotá D.C, diciembre de 2015

Para mi mamá la mujer que siempre me ha reforzado el concepto de ser humano con su manera de ser y con todo su amor, por ser mi modelo a seguir y guiarme por el camino más adecuado para alcanzar mi proyecto de vida.

A mis padres que han estado a mi lado siempre, apoyándome y patrocinándome en este sueño de ser profesional.

A mi hermana, porque gracias a ella soy la persona que soy, y no imagino que sería de mi si ella no hubiese estado siempre a mi lado para ayudarme a comprender el mundo de la manera en que lo hago.

- Hugo Sanchez

AGRADECIMIENTOS

A mi familia quienes me han brindado su apoyo incondicional a lo largo de este proceso y me han motivado a seguir adelante en los momentos más difíciles.

A los arquitectos e ingenieros expertos en SOA que me colaboraron en el desarrollo del caso de estudio anexo a esta investigación.

A mis profesores, especialmente al Ingeniero Gilberto Pedraza y al sociólogo Ignacio Hernández Molina, por brindarme su colaboración y su tiempo durante mi carrera de ingeniero en formación, logrando con este proyecto de grado un aporte de impacto para la comunidad académica.

TABLA DE CONTENIDO

	Pág.
INTRODUCCIÓN	18
1. PLANTEAMIENTO DEL PROBLEMA	20
1.1 DEFINICIÓN DEL PROBLEMA	20
1.2 JUSTIFICACIÓN	22
1.3 ALCANCES	23
1.4 LÍMITES	23
1.5 OBJETIVOS	24
1.5.1 General	24
1.5.2 Específicos	24
2. MARCO TEÓRICO	25
2.1 SOA SEGÚN LOS AUTORES ACADÉMICOS	25
2.1.1 ¿Qué es SOA?	26
2.1.1.1 Definición	26
2.1.1.2 Objetivos	28
2.1.1.3 Características	29
2.1.1.4 Beneficios	30
2.1.1.5 Instrumentos necesarios para su adopción	31
2.1.2 El esquema global de una SOA	31
2.1.2.1 Componentes del esquema global SOA	31
2.1.2.2 Componentes del Esquema global de la Organización	32
2.1.3 Políticas de SOA	33
2.1.3.1 Políticas y procesos durante la fase de diseño	34
2.1.3.2 Políticas y procesos durante la fase de ejecución	35
2.1.4 Infraestructura de servicios	35
2.1.4.1 Capacitación de servicios	35

2.1.4.2	Utilización de un bus empresarial (ESB)	36
2.1.4.3	La capa de mediación de servicios	36
2.1.4.4	Virtualización de los servicios	37
2.1.5	Composición	39
2.1.5.1	¿Qué es la composición?	39
2.1.5.2	Gestión de procesos de negocio (BPM)	39
2.1.6	Gestión de la evolución de la SOA	41
2.1.6.1	Medición de la eficacia de SOA	41
2.2	GOBIERNO SOA SEGÚN LA INDUSTRIA	42
2.2.1	IBM	42
2.2.1.1	La necesidad de Gobierno y administración	42
2.2.1.2	Proveer un servicio sin un Gobierno adecuado (Situación Hipotética)	43
2.2.1.3	Consumir un servicio (Situación Hipotética)	44
2.2.1.4	¿Qué es el Gobierno SOA?	45
2.2.1.5	Ciclo de vida del Gobierno y metodología	46
2.2.1.6	Definición del servicio	48
2.2.1.7	Control de versiones del servicio	49
2.2.1.8	Migración del servicio	50
2.2.1.9	Modelo de mensajes del servicio	51
2.2.1.10	Supervisión del servicio	52
2.2.1.11	Propiedad del servicio	53
2.2.1.12	Pruebas del servicio	54
2.2.1.13	Seguridad del servicio	55
2.2.2	ORACLE	56
2.2.2.1	Mejores Prácticas SOA, Estándares y Gobernabilidad	56
2.2.2.2	Estándares	58
2.2.2.3	Mensajería	58
2.2.2.4	Metadata para la Descripción y Descubrimiento de los Servicios	60

2.2.2.5	Servicios de Implementación	61
2.2.2.6	Mejores prácticas SOA	62
2.2.2.7	Gobernabilidad	62
2.2.2.8	Principios de Gobernabilidad	65
3.	METODOLOGÍA. ESTUDIO DE MAPEO SISTEMÁTICO	67
3.1	PREGUNTAS DE INVESTIGACIÓN	68
3.2	SELECCIÓN DE BIBLIOTECAS DIGITALES Y DEFINICIÓN DE LA ESTRATEGIA DE BÚSQUEDA	69
3.3	SELECCIÓN DE ESTUDIOS	71
3.4	EXTRACCIÓN DE DATOS Y SÍNTESIS	73
4.	RESULTADOS	75
4.1	PREGUNTA DE INVESTIGACIÓN 1	75
4.1.1	Selección del personal	75
4.1.2	Conformación del equipo de trabajo	75
4.1.3	Capacitar al equipo de trabajo	77
4.1.4	Seleccionar un conjunto limitado de Principios SOA	77
4.1.5	Establecer un conjunto de políticas	77
4.1.6	Plan de Comunicación	78
4.1.7	Plan de Capacitación	79
4.1.8	Establecimiento de comités	80
4.1.9	Establecimiento de Procesos	81
4.1.10	Diseño del ciclo de vida de los servicios	81
4.1.11	Diseño del Portafolio/Repositorio de Servicios	83
4.1.11.1	Trabajando con el Portafolio/Repositorio de servicios	84
4.1.11.2	Vinculación entre consumidores y servicios	86
4.1.11.3	Registrando servicios en el Portafolio/Repositorio	87
4.2	PREGUNTA DE INVESTIGACIÓN 2	88
4.2.1	Clasificación de las características de una herramienta adecuada para el Gobierno SOA	88

4.2.2	Características deseables en una herramienta adecuada para el Gobierno SOA	90
4.3	AMENAZAS A LA VALIDEZ	92
4.3.1	Validez de conclusión	93
4.3.2	Validez del constructo	93
4.3.3	Validez interna	93
4.3.4	Validez externa	93
4.4	DISCUSIÓN	94
5.	CONCLUSIONES Y TRABAJO FUTURO	96
5.1	CONCLUSIONES	96
5.2	TRABAJO FUTURO	97
	BIBLIOGRAFÍA	98
	ANEXOS	104

LISTA DE TABLAS

	Pág.
Tabla 1. Términos y sinónimos usados para componer la cadena de búsqueda	70
Tabla 2. Categorías de clasificación de los estudios	74
Tabla 3. Galaxy versus G – REG	117
Tabla 4. Galaxy versus G – REG – Evaluación de criterios comerciales	121

LISTA DE FIGURAS

	Pág.
Figura 1. Ejemplo de servicio virtual	37
Figura 2. Ciclo de vida de la SOA	47
Figura 3. Resumen de la arquitectura WS-Security en OracleAS	60
Figura 4. Annotated Java Class for Web Services	62
Figura 5. Estudio de mapeo sistemático	67
Figura 6. Ejemplo de servicio virtual	86
Figura 7. Accediendo al Galaxy desde una aplicación tipo Spring	107
Figura 8. Galaxy Query Language (GQL)	108
Figura 9. Mensaje de error de Galaxy	110
Figura 10. Ventana de búsqueda avanzada	113
Figura 11. Consola de G-Reg (Cuadro de mando)	115
Figura 12. Interfaz gráfica Membrane	120

LISTA DE ANEXOS

	Pág.
Anexo A. Implementación de las estrategias en el caso de estudio	104
Anexo B. Evaluación de plataformas open source para el Gobierno SOA	105
Anexo C. MULE – GALAXY	106
Anexo D. WSO2 - G-REG	112
Anexo E. Resultados de evaluación de plataformas open source para el Gobierno SOA (grafica)	116
Anexo F. PREDIC8 – MEMBRANE	117
Anexo G. Criterios comerciales	120

GLOSARIO

ARQUITECTURA DE SOFTWARE: es el proceso de definición de una solución estructurada que cumple con todos los requisitos técnicos y operativos, al tiempo que optimiza la calidad común atributos tales como el rendimiento, la seguridad y capacidad de administración.¹

ARQUITECTURAS ORIENTADAS A SERVICIOS (SOA): es una arquitectura para la construcción de aplicaciones de software que utilizan los servicios disponibles en una red como Internet. Promueve la articulación flexible entre los componentes de software para que puedan ser reutilizados. Las aplicaciones en SOA son construidas en base a los servicios².

ESTRATEGIA: la importancia de las estrategias de negocio radica en sus vinculaciones directas con los resultados empresariales. El concepto de estrategia se destaca a nivel de negocios debido a que una empresa necesita una dirección de expansión bien definida. Si los objetivos no satisfacen las necesidades empresariales se requieren reglas de decisión adicionales para que la empresa tenga un crecimiento ordenado y rentable³.

ESTUDIO DE MAPEO SISTEMÁTICO: Petersen et al define un estudio de mapeo sistemático como un estudio secundario que tiene como objetivo construir un esquema de clasificación y estructurar un campo de interés de la ingeniería de software. Para llevar a cabo el estudio sugiere un proceso de cinco pasos que son

¹ MICROSOFT CORPORATION. Application Architecture Guide [Online]. 2009. Disponible en: <<https://msdn.microsoft.com/en-us/library/ff650706.aspx>>

² MAHMOUD, Qusay. Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI) [Online]. Oracle, 2005. Disponible en: <<http://www.oracle.com/technetwork/articles/javase/soa-142870.html>>

³ ANDREU, Rafael. RICART, Joan. VALOR, Josep. Estrategia y Sistemas de Información. Mc Graw Hill. 1996. p. 372.

los siguientes: definir las preguntas de investigación, realizar la búsqueda de los documentos pertinentes, seleccionar los estudios primarios, análisis de los resúmenes y extracción de palabras clave, extracción de datos, y por último mapear los estudios primarios seleccionados⁴.

GOBIERNO SOA: en general, Gobierno significa establecer y exigir cómo acuerda trabajar un grupo en equipo. Específicamente, el Gobierno es el establecimiento de cadenas de responsabilidad para facultar a las personas, la medición de la efectividad de los medidores de la adopción, las políticas para guiar la organización con el fin de que cumpla sus objetivos, los mecanismos de control para garantizar el cumplimiento de los mismos y la comunicación para mantener todas las partes informadas⁵.

IBM: es una empresa multinacional estadounidense de tecnología y consultoría con sede en Armonk, Nueva York. IBM fabrica y comercializa hardware y software para computadoras, y ofrece servicios de infraestructura, alojamiento de Internet, y consultoría en una amplia gama de áreas relacionadas con la informática, desde computadoras centrales hasta nanotecnología⁶.

IEEE: es la mayor asociación profesional mundial dedicada a promover la innovación tecnológica y la excelencia en beneficio de la humanidad. IEEE y sus miembros inspiran una comunidad global a través de las publicaciones de IEEE altamente citados, conferencias, estándares de tecnología y las actividades profesionales y educativas⁷.

⁴ PETERSEN, K. FELDT, R. Systematic Mapping Studies in Software engineering. Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. 2008. p. 8.

⁵ WOHLIN, C. Experimentation in software engineering: an introduction. Springer. Vol. 6. 2000. p. 11.

⁶ RESEARCH IBM. Nanotechnology & Nanoscience [Online]. 2013. Disponible en: <<http://www.research.ibm.com/index.shtml>>

⁷ IEEE. Advancing Technology for Humanity [Online]. 2015. Disponible en: <<https://www.ieee.org/about/index.html>>

ORACLE: es una compañía de software que desarrolla bases de datos (Oracle Database) y sistemas de gestión de bases de datos. Cuenta además, con herramientas propias de desarrollo para realizar aplicaciones, como Oracle Designer, Oracle JDeveloper y Oracle Developer Suite. Su actual consejero delegado es Larry Ellison⁸.

PORTAFOLIO DE SERVICIOS: el portafolio de servicios constituye la parte más tangible de la implantación de una estrategia SOA, ya que publica y recopila de forma ordenada los servicios de negocio y de infraestructura haciéndolos así disponibles para todos los sistemas de información de la organización los cuales pueden reutilizarlos en un determinado momento de la operación del negocio⁹.

⁸ WIKIPEDIA. Oracle Corp [Online]. 2015. Disponible en: <https://es.wikipedia.org/wiki/Oracle_Corporation>

⁹ AFSHAR, M. SOA Governance: Framework and Best Practices. Oracle. 2007. p. 5.

RESUMEN

En la actualidad los arquitectos de software no cuentan con unos estándares o herramientas aceptadas comúnmente por la industria para apoyar los procesos de Gobierno en una SOA a nivel de todo el ciclo de vida de la arquitectura. Esta situación se puede contextualizar desde dos perspectivas, la primera es que si bien existen manuales para implementar un Gobierno adecuado estos pertenecen a empresas de la industria que por lo general venden toda una suite de productos a precios que solo podría asumir una empresa grande y posiblemente no cualquier arquitecto entusiasta que quiera llevar a cabo una implementación de Gobierno SOA. La segunda es que incluso teniendo acceso a estas herramientas privadas es difícil acordar una estrategia de Gobierno definitiva que funcione para todas las empresas, por lo que hablar de un manual universal que explique la adecuada implementación de un Gobierno SOA podría ser algo pretencioso.

La presente investigación tiene como objetivo brindar unas estrategias para la adecuada conformación de un Gobierno en una implementación SOA que permita a cualquier arquitecto de software implementar un Gobierno paso a paso. Estas estrategias se han seleccionado consultando autores teóricos de SOA, documentación disponible de empresas líderes de la industria en productos SOA y diferentes opiniones y experiencias de la comunidad de desarrolladores que ha venido trabajando con algunas herramientas de Gobierno libres.

Teniendo en cuenta que se necesitaba consultar, extraer y sintetizar gran cantidad de información, se realizó un estudio de mapeo sistemático mediante el cual se respondieron unas preguntas de investigación, lo que permitió definir y validar las estrategias para la adecuada implementación de un Gobierno paso a paso. Estas estrategias a su vez conforman el resultado de la investigación y permitieron cumplir el objetivo principal del estudio.

Palabras Clave - Servicio, Arquitectura Orientada a Servicios, Gobierno SOA, Estrategias de Gobierno, Portafolio de Servicios, Registro, Repositorio, Estudio de Mapeo Sistemático.

INTRODUCCIÓN

Muchas organizaciones han optado por desarrollar proyectos utilizando una Arquitectura Orientada a Servicios (SOA) con el fin de mejorar los tiempos invertidos en las modificaciones de sus procesos de negocio, evolucionar los modelos de negocio, facilitar la integración de sistemas y aplicaciones de diferentes proveedores, etc. Sin embargo pretender desarrollar una SOA sin la implementación de un Gobierno adecuado, es equivalente a intentar construir un edificio sin cimentación, lo cual generará un ambiente de inestabilidad tan grande que afectará no solo a la organización implicada en dicho proceso, sino también a sus proveedores y clientes.

Una de las promesas de la Arquitectura Orientada a Servicios es crear servicios de alta calidad que puedan ser fácilmente consumidos a nivel interno y externo, especificando los niveles de servicios acordados (SLA). Sin embargo la mayoría de los proyectos SOA no cuentan con unos estándares o herramientas aceptadas comúnmente por la industria para apoyar los procesos de Gobierno en una SOA a nivel de todo el ciclo de vida de la arquitectura, generando un Gobierno insuficiente, siendo este activo la base para que la Arquitectura se desarrolle y evolucione.

Por esta razón, se planteó solucionar dicha problemática a través de un estudio de mapeo sistemático. Esta metodología de investigación consta esencialmente de cinco grandes pasos. Definir las preguntas de investigación, realizar la búsqueda de los documentos pertinentes, seleccionar los estudios primarios, análisis de los resúmenes y extracción de palabras clave, extracción de datos, y por último mapear los estudios primarios seleccionados. Este estudio tuvo como principal objetivo recolectar y extraer información de artículos de investigación actuales que realizaran propuestas sobre cuáles serían las mejores estrategias o tácticas para

llevar a cabo una implementación de Gobierno SOA exitosa, y como resultado de este estudio se respondieron unas preguntas de investigación. Estas respuestas son las que conforman el conjunto de estrategias que se validaron y se pusieron a prueba en un caso de estudio.

A partir de la extracción de información realizada, se determinó que un buen tema a profundizar seria el diseño y construcción del portafolio/repositorio de servicios, insumo de vital importancia en la implementación de un Gobierno que sea consecuente con las expectativas de los arquitectos de software en proceso de adoptar una SOA que evolucione y cumpla los objetivos trazados.

En el presente documento se puede encontrar un total de cinco secciones. En la primera se encuentra el planteamiento del problema compuesto por la definición del mismo, la justificación, los alcances, los límites y los objetivos de la investigación. En la segunda se encuentra toda la fundamentación teórica de la investigación, en la tercera se encuentra la metodología de investigación (el estudio de mapeo sistemático), en la cuarta los resultados obtenidos de la metodología de investigación y en la quinta se encuentran las conclusiones de la investigación y las propuestas para realizar trabajos futuros.

1. PLANTEAMIENTO DEL PROBLEMA

1.1 DEFINICIÓN DEL PROBLEMA

La Arquitectura Orientada a Servicios (SOA) es una estrategia que ha venido cobrando fuerza en la integración e interoperabilidad de aplicaciones y plataformas del mercado con el fin de consumirlas en la red de manera rápida, evitando que cada empresa tenga que invertir recursos en crearlas desde cero¹⁰. SOA aumenta el nivel de velocidad de respuesta ante un cambio de negocio y el nivel de coordinación que se requiere entre las empresas y la tecnología de la información, así como entre los departamentos y los equipos de IT. Esta coordinación es proporcionada por el Gobierno SOA, que abarca las tareas y procesos para especificar cómo se administran los servicios y las aplicaciones de la implementación SOA.

En general, gobernar significa establecer y definir cómo acuerda trabajar un grupo en equipo. En otras palabras, es el establecimiento de responsabilidades y de políticas para guiar la organización hacia el cumplimiento de sus objetivos. Es necesario además que el proceso de gobernar la implementación de una SOA genere sus propios mecanismos de control para demostrar el cumplimiento de dichas políticas, garantizando a su vez una comunicación efectiva entre todas las partes responsables de la adopción de una SOA.

Uno de los puntos más importantes para desarrollar adecuadamente un Gobierno SOA es la construcción, mantenimiento y evolución del Portafolio de Servicios. Muchas iniciativas de Gobierno SOA menosprecian el valor de este activo y no

¹⁰ ERL, T. SOA Governance Thomas Erl. Boston: Prentice Hall Professional Technical Reference. 2007. p. 261.

gestionan adecuadamente su información, ocasionando un desconocimiento de la importancia de cada servicio desarrollado por la Organización.

Sí el portafolio de servicios no existe o su información se encuentra desactualizada, las decisiones que se tomen sobre los diferentes servicios estarán soportadas únicamente por el conocimiento con que cuenten las personas que así lo decidan, generando una desarticulación de las necesidades reales que motivaron estas decisiones. Esto generaría un incremento en los costos ya sea por implementación de servicios ya desplegados en producción, o por la pérdida de la capacidad estratégica al tomar decisiones sin información confiable.

Cuando esto ocurre, aparecen por ejemplo sistemas de información que no terminan de evolucionar a una SOA, sistemas que no terminan de publicar servicios claramente identificados, o que no terminan de consumirlos. Aparecen áreas funcionales completas que parecen ajenas a la estrategia de adopción, y que continúan lanzando proyectos y nuevos sistemas de información como si la estrategia SOA no estuviera alineada con ellos.

Debido a la interoperabilidad entre los sistemas y la alta reutilización de los servicios que ofrece SOA, las decisiones que modifiquen el portafolio de servicios o los ecosistemas SOA tiene un impacto potencial en muchos segmentos del negocio. Cuando este impacto no se evalúa previamente, se pueden presentar consecuencias que pueden ir desde la interrupción de los procesos de negocio hasta el abandono en la implementación de la arquitectura SOA por pérdida de credibilidad en el paradigma.

Sin embargo, aún no se cuentan con estándares ni herramientas aceptadas comúnmente por la industria para apoyar estos procesos de gobierno a nivel de todo el ciclo de vida de la arquitectura, y es justamente este punto donde surge la necesidad en la que se basa esta investigación, ya que el área de Gobierno del

semillero de investigación SOA Research de la Universidad Piloto de Colombia tiene como principal objetivo proponer una serie de estrategias para la construcción, mantenimiento y evolución del Gobierno SOA y su correspondiente portafolio de servicios, dirigidas a los arquitectos de software que no disponen de los recursos económicos necesarios para adquirir los productos de las empresas líderes del mercado (IBM, Oracle, Mulesoft, etc.) permitiéndoles establecer los criterios adecuados para elegir las estrategias y las herramientas de Gobierno que se mejor se adapten a su implementación SOA y a sus necesidades de negocio en particular.

1.2 JUSTIFICACIÓN

El Gobierno es la columna vertebral de una SOA dado que especifica cómo se administran los servicios y las aplicaciones de una organización, por esta razón surge la necesidad de formular un conjunto de estrategias que ayuden al arquitecto de software a analizar de manera óptima y eficiente la manera de implementar un Gobierno SOA sin tener que acudir al uso de las herramientas ofrecidas por las compañías líderes en el mercado.

Esto generaría gran impacto para la comunidad, puesto que se está diseñando una solución de bajo costo a partir de herramientas libres disponibles para toda la comunidad de arquitectos de software, razón por la cual no se tendría la necesidad de pagar costosas licencias que generen más gastos para una compañía en proceso de adopción, por el contrario se generarían mayores ganancias.

Incorporar esta investigación a la propuesta metodológica del semillero mejoraría la efectividad de la adopción SOA y un mejor resultado en inversión de recursos para cualquier entidad, ya que se promueve el uso de herramientas libres disponibles para cualquier persona que tenga la necesidad de realizar una

construcción, mantenimiento y evolución del Gobierno y su correspondiente portafolio/repositorio de servicios en una arquitectura orientada a servicios.

1.3 ALCANCES

Para este proyecto de investigación se plantea recopilar una serie de estrategias para la conformación de un Gobierno SOA y su portafolio de servicios, obtenidas a través de un estudio de mapeo sistemático que permita extraer la información necesaria para validar dichas estrategias teniendo en cuenta algunos trabajos previos que han realizado autores SOA, la comunidad de desarrollo y empresas líderes del mercado como Oracle e IBM.

El producto resultante de esta investigación podrá ser usado por arquitectos o ingenieros de software con o sin experiencia en SOA que requieran implantar un Gobierno y un portafolio de servicios en una arquitectura orientada a servicios.

1.4 LÍMITES

Esta investigación se centrará en plantear una serie de estrategias para la conformación de un Gobierno SOA y su portafolio de servicios, las cuales se limitan al criterio de efectividad que un arquitecto de software considere que tienen para su implementación SOA en particular. Por esa misma razón las estrategias se plantean bajo condiciones ideales, para cualquier tipo de herramienta, permitiéndole así a cualquier arquitecto o ingeniero de software conocer los lineamientos para poner en práctica dichas estrategias de forma adecuada.

Se planteará un conjunto de estrategias para la conformación de un Gobierno SOA y su portafolio de servicios, por tanto no se tocarán temas propios de proyectos de software referentes a desarrollos, programas o codificaciones. No se desarrollarán trabajos de códigos, ni referentes a desarrollos de programas.

1.5 OBJETIVOS

1.5.1 General. Brindar una serie de estrategias para la conformación de un Gobierno y su portafolio de servicios en una implementación SOA que puedan ser implementadas por cualquier arquitecto o ingeniero de software (sin la necesidad de adquirir una plataforma propietaria de un proveedor específico) para que alcancen el cumplimiento de sus objetivos.

1.5.2 Específicos

- Realizar un estudio de mapeo sistemático que permita extraer la información necesaria para validar las estrategias propuestas teniendo en cuenta algunos trabajos previos que han realizado autores SOA, la comunidad de desarrollo y algunas empresas líderes del mercado.

- Determinar los criterios de selección más importantes para los arquitectos e ingenieros de software al momento de decidirse por una herramienta de gobierno que permita la gestión adecuada del portafolio de servicios teniendo en cuenta las necesidades específicas de negocio que requiera su implementación SOA.

2. MARCO TEÓRICO

2.1 SOA SEGÚN LOS AUTORES ACADÉMICOS

Desde que la necesidad por el manejo de la información surgió, los ingenieros informáticos nos hemos visto llamados a brindar soluciones adaptativas para cada cliente que solucionen sus problemas de control y administración de sus procesos personales o empresariales. La programación precisamente ha sido la herramienta de la que nos hemos valido para brindar soluciones a la medida de cada cliente o necesidad, implementando día a día mejores herramientas mucho más fáciles de usar y aprovechando mejor los recursos para llevar a cabo dicha tarea.

Las Tecnologías de la Información (IT) han tomado un papel vital en el trabajo de los desarrolladores de software, ya que por muchos años nos han venido acompañando y sugiriendo modelos de negocio que en su momento funcionaron, y de hecho aún funcionan, pero que al día de hoy observamos algunas deficiencias en cuanto a manejo eficiente de recursos, de acceso por parte de los clientes a las herramientas realizadas o simplemente de adaptabilidad a las necesidades cambiantes y cada vez más exigentes de las organizaciones.

Hoy en día las organizaciones y empresas de desarrollo de software tienen la necesidad de adaptarse a constantes cambios en el mercado y en las exigencias de los consumidores, por tanto, deben ser capaces de adaptar sus procesos a unas condiciones dinámicas como única forma de mantener y mejorar su competitividad y sus ingresos. A sí mismo, los sistemas que dan soporte a esos procesos deben poderse adaptar fácilmente para responder de forma ágil a las necesidades del negocio. El estado del arte actual de la tecnología corresponde al reto de SOA, y son justamente estas innumerables dificultades presentes en los

modelos de negocio de IT las que el modelo de Arquitecturas orientadas a servicios entra a atacar con agilidad, reutilización, composición y flexibilidad en el manejo de los procesos.

2.1.1 ¿Qué es SOA?

2.1.1.1 Definición. La arquitectura orientada a servicios (SOA) adquiere diferentes significados de acuerdo al rol de los autores de la adopción del modelo a su negocio. Para un director de información SOA representa reducción de costos y el aumento de ganancias, para un ejecutivo de negocio es un conjunto de servicios cuya finalidad radica en adaptarse a nuevas necesidades, para un arquitecto de IT significa integrar sistemas con un acoplamiento bajo y reutilizables, para un arquitecto de empresa corresponde a la aplicación de dinámicas interoperables que reduce de manera drástica la complejidad y la rigidez deteniendo la entropía y finalmente para un desarrollador SOA representa un paradigma de programación normalmente representado en Web services, siempre teniendo en cuenta que una SOA se puede realizar como un Web service, pero no todos los Web services son una SOA.

Es importante aclarar que la esencia de SOA consiste en gran medida en no limitarse a una sola de estas perspectivas, si no en encontrar puntos de convergencia cuya finalidad sea facilitar su adopción. Después de varias revisiones bibliográficas resulta bastante apropiado decir que SOA se define como un medio arquitectónico, un paradigma donde todo cobra forma de servicios. Para este esquema de arquitectura es irrelevante donde están ubicados los servicios o su funcionamiento interno, lo que realmente es importante es la función que cumplan.

“En una SOA, los servicios se encuentran en un repositorio denominado registro, se ensamblan mediante las llamadas aplicaciones compuestas y el plano que le sirve de guía es lo que se conoce como esquema global de la SOA”¹¹

- **¿Qué es un servicio?** Un servicio es el elemento más básico de SOA cuya función principal radica en proporcionar capacidades al consumidor, contando con determinados métodos para invocarlos cuando se necesiten. Los servicios se componen de una interfaz, que determina la funcionalidad que un servicio proporciona y la implementación, que determina como se proporciona dicha funcionalidad. Se dice que son opacos dado lo irrelevante que es su funcionamiento interno, ya que su importancia radica en la función que cumple.

- **Características de un servicio.** Los servicios tienen diferentes características dependiendo en gran medida de la perspectiva para abordar SOA que se tenga en mente, sin embargo son comunes entre ellas:

- Sin estado. Los servicios no recuerdan lo último que se les pidió, ni les importa lo que les pidan a continuación, siendo independientes entre sí.
- Visible. Deben ser visibles en su totalidad en el registro/repositorio por los consumidores de servicios.
- Auto-Descripción. Al poseer una descripción de sí mismo se expone toda la información del servicio sin necesidad de que tenga que ser entendido por el consumidor.
- Composición. Los servicios también se componen de otros servicios.

¹¹ MATSUMURA Mike, BRULE Bjorn & SHAH Jingles. Adopción de SOA para dummies. Indiana: Wiley Publishing Inc., 2009. p. 24.

- **Bajo Acoplamiento.** Aunque es imposible un desacoplamiento total de los servicios cuanto menos dependiente sean, mejor será la adopción de SOA.
- **Regido por políticas.** Las políticas definen los acuerdos de nivel de servicio (SLA), la forma para invocar y usar adecuadamente un servicio, entre otros.
- **Ubicación, lenguaje y protocolo independientes.** Los servicios funciona y están disponibles sin importar el lugar donde se encuentren, la plataforma donde se realizaron o el tipo de usuario que los invoque.
- **Alta granularidad.** Reduce la complejidad para desarrolladores limitando los pasos necesarios para cumplir una función determinados reduciendo los recursos implementados.
- **Asíncrono.** Aumenta la escalabilidad del sistema con mensajería. Emite una solicitud de servicio y continúa cuando el proveedor emite una respuesta.

2.1.1.2 Objetivos. Las soluciones de tipo SOA han sido creadas para satisfacer los objetivos de negocio los cuales incluyen poner a disposición de los clientes soluciones combinando componentes, servicios y procesos en una única solución a la medida de las necesidades particulares. Para esto se hace uso de elementos reutilizables como interfaces y estructuras para componer nuevas interfaces de usuario que contemplan servicios y procesos disminuyendo la complejidad y la rigidez.

Otras de las promesas de SOA incluyen la facilidad y flexibilidad de integración, reduciendo costos de implementación para maximizar el ROI (Retorno de Inversión), la reducción de plazos de entrega y de los riesgos presentes en el desarrollo del proyecto, la innovación de servicios

permanente para los clientes y una adaptación ágil ante cambios incluyendo la respuesta anticipada a la competitividad¹².

2.1.1.3 Características. SOA posee un gran número de características, entre las cuales están:

- Granularidad gruesa, que selecciona por preferencia los componentes de mayor tamaño (denominados servicios).
- La importancia que tienen en sí las funciones de las interfaces/implementaciones y no como realicen dichas funciones.
- La declaración de las obligaciones por parte del proveedor y del consumidor mediante la estipulación de contratos.
- Acoplamiento débil que permite servicios más flexibles y menos dependientes unos de otros.
- Reutilización. SOA tiene claro que es mucho más rápido y recursivo generar nuevas aplicaciones a partir de piezas prefabricadas que escribir “de ceros” todas las funciones requeridas. Estas aplicaciones también cambian a medida que los objetivos y se ajustan durante la marcha del proyecto.
- Implementación de BPM (Gestión de procesos de negocio) para sus procesos.

¹² LEWIS, Grace. SMITH, A. Dennis, B. KONTOGIANNIS, Kostas. A Research Agenda for Service-Oriented Architecture (SOA): Maintenance and Evolution of Service-Oriented Systems. Software Engineering Institute. 2010. p. 27.

- SOA hace que se compartan responsabilidades, no pretende hacer cumplir nuevas responsabilidades centralizadas que generen resistencia a la adopción del SOA.
- Madurez. Los servicios al igual que la adopción de SOA en una organización poseen diferentes niveles de madurez. SOA no acumula beneficios con solo implementarse, si no que va generando cambios en la medida que su proceso de maduración sea mayor.

2.1.1.4 Beneficios. Las empresas que se manejan por IT poseen grandes inconvenientes que SOA de forma asertiva soluciona aplicando todo el conjunto de características que la describen. Cuando IT presenta problemas de integración, del gasto de tiempo en responder a requerimientos cambiantes y de imposibilidad de configurar procesos de negocio como se necesita, SOA responde con una manera más rápida y flexible de reconfiguración de procesos de negocio y actividades.

Cuando IT presenta costos de mantenimiento elevados y dificultad en obtener un ROI que permita realizar actualizaciones, SOA responde con una disminución de costos operacionales de tecnología de información. “SOA tiene bastante claridad en que el atributo más importante que una organización puede tener es la flexibilidad, por este motivo es justamente la flexibilidad el motor que le da a SOA su razón de ser”¹³.

Otros beneficios adicionales de la adopción de SOA en las organizaciones es la persecución de iniciativas de negocio, la valorización y modernización de aplicaciones estratégicas, la disminución del costo de aplicaciones e infraestructura, la búsqueda de reutilización para llegar más rápidamente al

¹³ MICROSOFT CORPORATION. Software Architecture and Design [Online]. 2012. Disponible en: <msdn.microsoft.com/en-us/library/ee658093.aspx>

mercado competitivo, el aumento de ingresos, la reducción de los ciclo de los procesos de negocio y del tiempo y costo de la integración de diferentes servicios.

2.1.1.5 Instrumentos necesarios para su adopción. La adopción de SOA se vale de los siguientes instrumentos básicos para poder generar resultados significativos y notables a largo plazo en una organización de negocio. En primer lugar es de vital importancia adoptar una actitud mental dirigida a que son los servicios los que finalmente satisfacen al cliente, mediante una reforma a la metodología de trabajo que implica decir “No más aplicaciones”, en adelante se trabajaran procesos estructurados de proyectos diferentes (ciclos de vida). En segundo lugar se necesita apaciguar las “guerra de clanes” entre las personas que trabajan dentro de una organización. El último instrumento es la tecnología en forma de herramientas centradas en la empresa para monitorizar la actividad empresarial y modelar o analizar procesos y flujos de negocio. “Sin embargo, la tecnología solo forma una pequeña parte de SOA siempre que los anteriores instrumentos mencionados estén presentes en la iniciativa de la adopción”¹⁴.

2.1.2 El esquema global de una SOA

2.1.2.1 Componentes del esquema global SOA. El esquema global de SOA representa el plano de construcción del modelo. En él se muestra el objetivo completo a cumplir. Atendiendo a una de las características principales de SOA estos objetivos son sensibles a ser ajustados sobre la marcha.

Adicional el plano debe tener una lista con:

- Servicios de negocio.

¹⁴ BARRAZA, Fernando. Modelado y Diseño de Arquitectura de Software [Online]. 2012. Disponible en: <http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:s2_conceptosdemodelado.pdf>

- Requisitos para la descripción de los servicios.
- Métricas de rendimiento de los servicios.
- Estándares de interoperabilidad.
- Esquemas de datos.
- Políticas.
- Requisitos de clasificación y localización de los servicios.
- Diseño de la infraestructura de la SOA. Se representa con un mapa que contiene el software y el hardware necesario para la implementación de SOA.
- Plan de acción. Un paso a paso para ejecutar adecuadamente el esquema global disminuyendo los riesgos y ajustándose durante todo el proceso.

2.1.2.2 Componentes del Esquema global de la Organización. Haciendo parte del esquema global de SOA, el esquema global de la organización muestra la configuración que adoptara la organización definitiva de la SOA. “También Ayuda a reestructurar el departamento informático que anteriormente funcionaba para IT y se movía de acuerdo a sus objetivos”¹⁵.

Entonces, se deben contemplar los siguientes puntos:

- Evaluación de competencias. Aquí nos preguntamos si se cuenta con las competencias necesarias para adoptar una SOA.

¹⁵ UNIVERSIDAD DE SEVILLA. Introducción a la arquitectura del Software [Online]. 2013. Disponible en: <www.lsi.us.es/docencia/get.php?id=6496>

- Estructura de la organización. Se busca mejorar la distribución de responsabilidades entre el proveedor y el consumidor.
- Cuerpo de gobierno. Define las políticas y procesos en la adopción de SOA.
- Incentivos al comportamiento. Se usan para fomentar SOA dando compensaciones e incentivos profesionales de acuerdo al comportamiento con el cambio de paradigma.
- Roles y responsabilidades. Se ajustan las responsabilidades y los puestos generando compatibilidad con SOA.
- Modelo compartido de financiación de infraestructuras. Se identifican las entidades que pagan por los servicios ofrecidos y los cambios realizados.
- Métricas compartidas. Selección de herramientas de medición que brinden información permanentemente sobre el estado de la SOA.
- Sistema de ciclo de vida. Conjunto de pasos para diseñar, desplegar, mantener y retirar servicios.
- Plan de acción para el desarrollo de la organización. Estrategia para avanzar paso a paso a un esquema global de la organización. Es de vital importancia que cada proyecto represente un ROI (Retorno de la inversión) motivando la instauración de nuevos proyectos.

2.1.3 Políticas de SOA. El centro de competencia SOA es el órgano de gobierno que crea y aplica las políticas de SOA. Está conformado por representantes de cada clan que se vean afectados por los planes de SOA. Estos grupos entienden y aceptan las implicaciones del esquema global de SOA en su desempeño

profesional. Las políticas se expresan siguiendo un formato declarativo y se definen, cambian o eliminan de acuerdo a las necesidades. El éxito del cumplimiento de las políticas consiste en aplicarlas de forma activa y automática realizando un seguimiento con puntos de control (PEP).

2.1.3.1 Políticas y procesos durante la fase de diseño

- Interoperabilidad. Se especifica un medio uniforme a través de la ratificación de un conjunto de estándares.
- Capacidad de descubrimiento. Conjunto de elementos que permiten navegar a través de los servicios por medio de políticas.
- Seguridad. Se proporciona un medio uniforme que brinde seguridad a todos los servicios de la SOA a través de la implementación de políticas.
- Unicidad. Garantiza que los servicios no repiten nombres de otros servicios existentes.
- Conformidad con la Interfaz. Se precisa un medio uniforme para invocar los servicios.
- Conformidad con el formato de datos. Se buscan esquemas que garanticen que un servicio puede usar un campo de dirección empleado por otro, pese a posibles cambios en los sistemas de almacenamiento de datos.
- Métricas. Información estadística y generación de informes del diseño de servicios.

2.1.3.2 Políticas y procesos durante la fase de ejecución

- Acuerdos de nivel de servicio. Los proveedores y los consumidores establecen acuerdos en el rendimiento esperado, etc.
- Autenticación. Los proveedores y los consumidores establecen además la forma en la que los consumidores de servicios se identifican y el sistema que implementaran para llevar a cabo dicha tarea.
- Autorización. Métodos para determinar si un consumidor está autorizado a invocar determinado servicio.
- Encriptado. Busca enmascarar el contenido.
- Firmas. Garantiza el envío de mensajes entre proveedores y consumidores íntegros y válidos.
- Alertas y notificaciones. Bajo qué condiciones las alarmas y las notificaciones serán activadas dentro del proceso.
- Métricas. Estos indicadores de rendimiento (KPI - *Key Performance Indicators*) resultan claves para la ejecución de toma de decisiones de la organización mediante políticas.

2.1.4 Infraestructura de servicios

2.1.4.1 Capacitación de servicios. SOA defiende como principio la idea de que antes de crear nuevos servicios, es necesario verificar que no existan, favoreciendo la optimización del tiempo, la reutilización y la disminución de costos. Se hace necesario entonces implementar aplicaciones que existan y mejor aún,

que ya han sido verificadas e implementadas anteriormente. Esto genera confianza en los consumidores. Conservar lo existente es más rápido y económico, por eso no se recomienda en la adopción de SOA desechar los sistemas existentes por orientados a servicios, es mejor crear una nueva capa compuesta de servicios.

Las aplicaciones existentes disponibles a la reutilización se clasifican en aplicaciones desarrolladas Internamente y aplicaciones comercializadas. Las aplicaciones desarrolladas Internamente son las que los empleados de la empresa han desarrollado por si mismos para la propia empresa, valiéndose de sus propios medios para brindar soluciones a los problemas de información que afronta la organización en un momento determinado. Las aplicaciones comercializadas corresponden a herramientas creadas por terceros (ERP, CRM, entre otros) y personalizadas para dar solución a las necesidades de la empresa objetivo.

2.1.4.2 Utilización de un Bus Empresarial (ESB). Si la aplicación que se está intentando habilitar proporciona una interfaz para conectarla a otros sistemas podemos hacer uso de un Bus empresarial, el cual posee compatibilidad con varios protocolos, compatibilidad con diferentes patrones de comunicación, como el petición/respuesta, compatibilidad con diferentes formatos de mensajes traduciendo XML a un lenguaje comprensible para aplicaciones desde o hacia XML y finalmente adaptadores o wrappers que permiten conectarse a interfaces de aplicaciones antiguas independiente de protocolos y formatos. En este último caso se exponen los programas internos como servicios aunque cabe resaltar que si se le desea incorporar a la infraestructura de mediación de servicios, es necesario asegurarse que incluye de serie un medio sencillo para la virtualización.

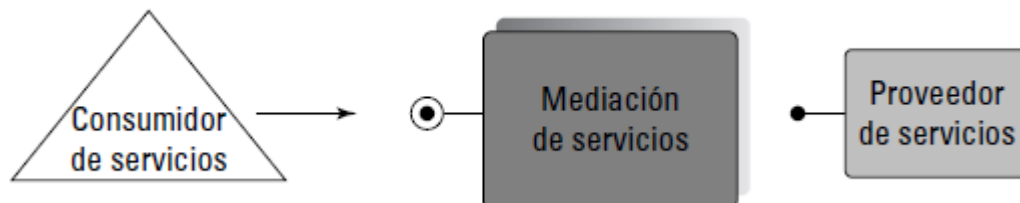
2.1.4.3 La capa de mediación de servicios. La capa de mediación de servicios promueve el acoplamiento débil separando proveedores y consumidores,

mejorando su interoperabilidad y planteando una infraestructura centralizada para que los desarrolladores se centren en la creación de la lógica del negocio. “En este capa se ponen las PEP (Punto de aplicación de políticas), permitiendo uniformidad e independencia sin importar donde se encuentren el proveedor o el consumidor”¹⁶.

Esta capa maneja diversidad de protocolos y formatos diferentes para mensajes, siendo el lugar ideal para aplicar políticas de vinculación y acuerdos en tiempos de ejecución. Lo interesante de la capa surge de la unión de mediación de servicios y gobierno en tiempo de ejecución, ya que esto favorece la reutilización de servicios y la introducción de cambios más fácilmente.

2.1.4.4 Virtualización de los servicios. Para poder llevar a cabo la mediación de servicios es necesario plantearlo como un servicio virtual de la siguiente forma:

Figura 1. Ejemplo de servicio virtual



Fuente: MATSUMURA Mike, BRULE Bjorn & SHAH Jingles¹⁷.

En gran medida, el proceso de virtualización promueve el acoplamiento débil, al generar independencia de la ubicación moviendo la implementación del servicio sin interrumpir a los consumidores, independencia del transporte al satisfacer diferentes formas de intercambiar información con mayores oportunidades de

¹⁶ ERL, T. Service Oriented Architecture - Concepts, Technology, and Design. Boston: Prentice Hall Professional Technical Reference. 2005. p. 129.

¹⁷ MATSUMURA, op. cit, p. 28.

reutilización y finalmente la independencia del mensaje, la cual transforma mensajes y los adecua a los diferentes tipos de consumidores/proveedores.

▪ **Requisitos Operativos.** Con la finalidad de realizar la virtualización de los servicios se necesita cumplir con los siguientes requisitos operativos:

- Validación de mensajes. Se verifica que los bloques XML se encuentren bien formados.
- Autenticación y autorización. Hace referencia a la autorización que tiene un consumidor para invocar determinado servicio.
- Encriptado de mensajes y firma. Desencripta los mensajes y realiza verificación de credenciales y firmas.
- Disponibilidad alternativa (failover) y balanceo de cargas. Capacidad para responder a la carga de transacciones.
- Enrutado de mensajes. Envía mensajes a diferentes implementaciones del servicio.
- Monitorización de los acuerdos de nivel de servicio (SLA). Garantiza el estado y el rendimiento del servicio compatible con cualquier proveedor con el fin de mejorar la adopción de SOA.

Dentro de los procesos de virtualización se encuentran los de utilización de intermediarios/pasarelas de servicios y los de utilización de equipos especializados en SOA, como los appliances. Los de utilización de intermediarios se centran en la virtualización de los servicios y tienen menos capacidad de

extensión y programación que los ESB. Su configuración, manejo y mantenimiento es más sencillo para los administradores de SOA.

Los de implementación de appliances funcionan como intermediarios entre servicios que incluyen Software y Hardware que procesa grandes cargas de XML y se distinguen por su instalación inmediata. Posee compatibilidad de estándares de seguridad con protección Incorporada y está capacitado para realizar funciones de compresión y descompresión de mensajes, entre otras. Cabe mencionar que los appliances aportan seguridad y rendimiento en perímetros de red, funcionando aun cuando los consumidores o proveedores están fuera de las fronteras confiables.

2.1.5 Composición

2.1.5.1 ¿Qué es la composición? SOA sin lugar a dudas se presenta como la nueva era de la composición, la cual, tiene como promesa la *agilidad*. Una característica de los sistemas ágiles consiste en la posibilidad de adaptar los servicios a nuevas necesidades de negocio que cambien durante el proceso de adopción de la SOA. Para esto es necesario contar con el adecuado conjunto de servicios y tecnología impulsada desde una perspectiva de negocio.

2.1.5.2 Gestión de procesos de negocio (BPM). Gestión de procesos de negocio (Business Process Management) hace referencia a una metodología corporativa cuyo objetivo es principalmente promocionar implementaciones automatizadas de procesos de negocio, mejorando el desempeño (Eficiencia y Eficacia) de la Organización a través de la gestión de los procesos de negocio, los cuales se deben diseñar, modelar, organizar, documentar y optimizar de forma continua.

Dentro de las principales ventajas del modelado BPM encontramos el entendimiento, visibilidad y control de los procesos de negocio de una organización. Un proceso de negocio representa básicamente una serie discreta de actividades o pasos de tareas que pueden incluir, personas, aplicativos, eventos de negocio y organizaciones¹⁸.

Cuando se decide combinar BPM con SOA se obtienen muchas ventajas dentro de la adopción:

- Un proceso de negocio se divide en fases que se pueden ver como servicios reutilizables.
- SOA garantiza rápida implementación de procesos usando servicios. Cada vez es más rápido automatizar procesos de negocio.
- Modificar un proceso requiere poca programación.
- SOA y BPM proporcionan puntos concretos de monitorización. Las empresas se valen de esto para encontrar los aspectos por mejorar.
- BPM tiene un lenguaje común personal.
- Podemos generar la fusión de BPM y SOA apoyándonos en tecnologías compuestas que permiten crear aplicaciones conectando servicios a una interfaz sin programación usando partlets (web) y mashups (aplicaciones).

¹⁸ DURVASULA, Surekha. GUTTMANN, Martin. KUMAR, Ashok. LAMB, Jeffery. MITCHELL, Tom. ORAL, Burc. PAI, Yogish. SEDLACK, Tom. SHARMA, Harsh. SUNDARESAN, Sankar Ram. SOA Practitioners Guide. Part 3. Introduction to Services Lifecycle [Online]. 2006. Disponible en: <<http://www.soablueprint.com/whitepapers/SOAPGPart3.pdf>>

2.1.6 Gestión de la evolución de la SOA. En el proceso de gestionar la evolución de la adopción SOA se debe asignar un proceso dedicado dentro del ciclo de vida, para gestionar el cambio de servicios y de procesos. En el proceso, se incluye una revisión de los contratos entre consumidores y proveedores para evaluar el impacto que genera cada cambio. Una vez se realiza esto se notifica a los consumidores activos cuando se vaya a producir un cambio inminente, o cuando haya una solicitud de cambio que deban aprobar. Un consejo clave para gestionar la evolución es el de introducir un atributo de “fecha de caducidad” en los contratos entre proveedores y consumidores.

Si el desarrollo soporta versiones simultáneas, se podrá disponer de más margen para intentar equilibrar la necesidad de cambiar las soluciones, y los deseos de estabilidad de las mismas.

2.1.6.1 Medición de la eficacia de SOA. SOA fracasa en el momento que no hay forma de medir su éxito. Las empresas en numerosas ocasiones únicamente compran soluciones marcadas como “Productos SOA” pero no miden su eficacia, faltando a una de los principales requisitos para la correcta adopción de SOA, el seguimiento y la búsqueda permanente de errores para adaptar servicios que los corrijan. Otra de las ideas que se presentan a la hora de intentar adoptar una SOA en una organización es que se estas ven el modelo de madurez de capacidades (CMMI) como un enfoque de mejora de procesos. Los principales aspectos que cambian cuando se realiza un seguimiento y una adopción adecuada son que:

- Mejora la capacidad para responder a las necesidades de las empresas. Se reduce el tiempo para implementar soluciones.
- Acelera el tiempo de entrega. Encontramos ciclos de prueba reducidos, ganando notablemente más tiempo.

- Mejora la flexibilidad de aplicaciones. Se reduce el tiempo y la velocidad de movimiento del ciclo de vida.
- Mejora de los procesos de negocio. Aumenta considerablemente el número de procesos que pueden ser reconfigurados.
- Reduce del costo de vida de las aplicaciones. Se reducen los recursos para producir y mantener código, el número de defectos y de servicios que se crean.
- Reduce el costo de integración. Presenta una reducción significativa del costo de creación de interfaces e infraestructura.
- Mejora la productividad. Se reduce el tiempo necesario para crear un servicio, el costo y el tiempo de entrega.
- Reutiliza activos de SOA. Aumenta el número de servicios consumidos por múltiples consumidores.
- Plantea estrategias a largo plazo de apoyo a las aplicaciones. El número de funciones de múltiples aplicaciones son convertidas a servicios basados en el Modelo de Madurez de Integración de Servicios (SIMM).

2.2 GOBIERNO SOA SEGÚN LA INDUSTRIA

2.2.1 IBM

2.2.1.1 La necesidad de Gobierno y administración. La Arquitectura Orientada a Servicios (SOA) es una técnica impactante para el desarrollo de aplicaciones de software que mejor se alinean con los modelos comerciales. Sin embargo, SOA aumenta el nivel de cooperación y de coordinación que se requiere entre las

empresas y la tecnología de la información, así como entre los departamentos y los equipos de IT. Esta cooperación y coordinación es proporcionada por el Gobierno SOA, que abarca las tareas y los procesos para especificar y administrar cómo se soportan los servicios y las aplicaciones SOA.

2.2.1.2 Proveer un servicio sin un Gobierno adecuado (Situación Hipotética).

Supongamos que se desarrolla un servicio para convertir una suma monetaria de una moneda en otra moneda. Se necesita en lugares diferentes, uno de ellos en un programa de procesamiento de pedidos en el que se está trabajando, entonces se escribe un código como una función reutilizable que se puede invocar desde cualquier lugar en un programa. Luego se necesita en otro programa, entonces se ingresa el código en un Java.jar que se puede agregar a la clase de cualquier programa que lo necesita. Pero un problema con el servicio es que lleva mucho tiempo iniciarse dado que necesita inicializarse con tarifas de cambio de divisas.

Estos pasos llevan mucho tiempo de inicialización cada vez que necesita convertir una suma monetaria. Entonces, se aloja el conversor en el propio programa, se inicializa y siempre está ejecutado y a su vez, siempre, puede llamarse desde uno de los programas a través del API remoto. “Esta API se puede implementar como un servicio web HTTP sobre SOAP o simplemente como una interfaz remota de Enterprise JavaBeans (EJB) que soporta RMI sobre IIOP”¹⁹.

Lo que se tiene ahora es un servicio de conversión de divisas. No solo lo utilizan un par de programas, sino que algunos compañeros de trabajo del departamento se ven interesados en dicho servicio y comienzan a invocarlo desde otros programas. Supongamos ahora que desde hace mucho, existen programas en otros departamentos en la empresa que lo están usando y de los cuales no se tiene conocimiento. El conversor está ejecutándose de tal manera que los tiempos

¹⁹ LEWIS, op. cit, p. 33.

de respuesta son lentos, entonces surge la necesidad de comprar una máquina más poderosa para alojar el servicio.

Al cabo de un tiempo la máquina sufre un daño crítico y el problema se da a conocer porque otros empleados de la compañía han detectado que el conversor dejó de funcionar, y es entonces cuando se toma la decisión de deshabilitar el conversor. Los empleados de la compañía empiezan a realizar denuncias sobre la no operación de dicho programa y sobre la influencia que tiene el mismo en otros programas que lo necesitan para funcionar.

2.2.1.3 Consumir un servicio (Situación Hipotética). Consideremos otra situación dentro de la misma compañía. Un empleado de esa empresa trabaja con una aplicación de catálogos de productos, ya que otros usuarios de diferentes países quieren poder ver cuánto cuestan los productos en su moneda.

Dicho servicio se da a conocer dentro de la compañía, se prueba y efectivamente se observa que funciona, entonces se implementa en la aplicación sobre la que se tenía el anterior conversor. Todos los empleados de la compañía que se vieron afectados adoptan esta función y las ventas en el sitio web de la compañía aumentan de forma notable²⁰.

Entonces, un día cualquiera el sitio web de la empresa deja de funcionar, ya que no puede mostrar los precios en otras divisas. En ese momento se procede a acudir por soporte a la persona que tomó la decisión de implementar dicho servicio dentro del aplicativo, que a su vez necesitara soporte de la persona que aparentemente diseñó este servicio, y del cual no se tiene ninguna información.

En este caso se evidencia una clara falta de Gobierno efectivo. Tanto el proveedor del servicio como el consumidor son responsables de lo sucedido y es necesario

²⁰ MICROSOFT CORPORATION. 2012, op. cit.

entender cómo se pueden utilizar los servicios sin que se descontrolen de esa forma.

2.2.1.4 ¿Qué es el Gobierno SOA? En general, Gobierno significa establecer y exigir cómo acuerda trabajar un grupo en equipo. Específicamente, el Gobierno es el establecimiento de:

- Cadenas de responsabilidad para facultar a las personas.
- Medición de la efectividad de los calibradores.
- Políticas para guiar la organización con el fin de que cumpla sus objetivos.
- Mecanismos de control para garantizar el cumplimiento.
- Comunicación para mantener todas las partes informadas.

El Gobierno determina quién es responsable de tomar decisiones, qué decisiones deben tomarse y las políticas para tomar decisiones de forma consistente. A diferencia de la gestión, el Gobierno planea qué decisiones se necesitarán realizar, mientras que la administración es el proceso de tomar e implementar decisiones. El Gobierno establece políticas, mientras que la administración las respeta.

“El Gobierno SOA es una especialización del Gobierno de IT que coloca las decisiones clave de Gobierno de IT dentro del contexto del ciclo de vida de los componentes, los servicios y los procesos comerciales. Es la gestión efectiva de este ciclo de vida la meta clave del Gobierno SOA”²¹.

²¹ UNIVERSIDAD DE SEVILLA, op. cit.

El Gobierno de la IT es más amplio que el Gobierno SOA. La que el primero cubre todos los aspectos de IT, incluidas las cuestiones que afectan la SOA, como los modelos de datos y la seguridad, así como cuestiones más allá de SOA como el almacenamiento de datos y el soporte de escritorio. “El Gobierno SOA abarca aspectos del ciclo de vida del servicio como el planeamiento, publicaciones, descubrimientos, control de versiones, gestión y seguridad”²².

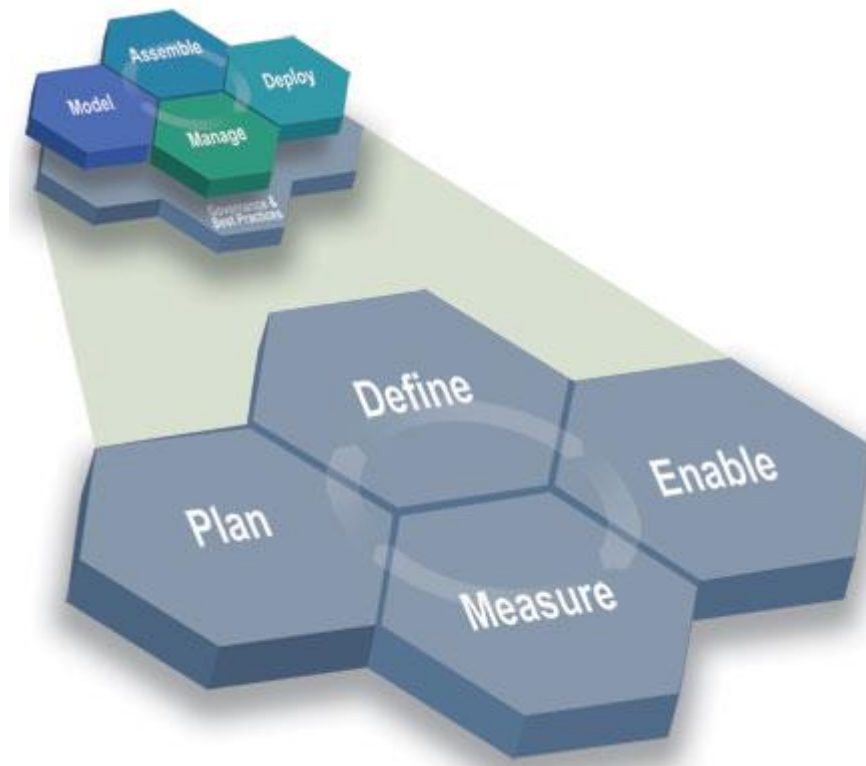
El Gobierno se hace más importante en la SOA que en la IT general. En la SOA, los consumidores del servicio y los proveedores del servicio ejecutan diferentes procesos, son desarrollados y gestionados por diferentes departamentos y requieren de mucha coordinación para trabajar juntos de forma exitosa. Para que la SOA tenga éxito, varias aplicaciones necesitan compartir servicios comunes, lo que significa que necesitan coordinar para que esos servicios sean comunes y reutilizables.

Dado que las empresas utilizan SOA para alinear mejor la IT con las empresas, se puede utilizar el Gobierno de la SOA para mejorar el Gobierno en general de la IT. La implementación del Gobierno SOA es clave para que las empresas puedan darse cuenta de los beneficios de la adopción SOA, de hecho, para que la SOA sea exitosa, el Gobierno técnico y comercial no es opcional, es una necesidad.

2.2.1.5 Ciclo de vida del Gobierno y metodología. El desarrollo del servicio sigue un ciclo de vida que IBM denomina el ciclo de vida de la SOA. El Gobierno SOA también sigue un ciclo de vida, el ciclo de vida del Gobierno SOA. Estos dos ciclos de vida funcionan juntos, se ejecutan juntos y se utilizan de forma conjunta para producir aplicaciones combinadas de la SOA y sus servicios. El ciclo de vida del Gobierno produce un modelo de Gobierno que se utiliza para gestionar el ciclo de vida de la SOA. Este ciclo se muestra en la siguiente ilustración:

²² SOA BLUEPRINT. About US, Strategic Use of IT [Online]. Disponible en: <http://www.soablueprint.com/about_us>

Figura 2. Ciclo de vida de la SOA



Fuente: SADTLER, Carla. HUBER, Philipp. YI, SangMin²³.

El Gobierno SOA de IBM y el Método de Administración (SGMM) es un proceso completo para llevar a cabo el ciclo de vida del Gobierno SOA, entonces dicho Gobierno puede aplicarse al ciclo de vida de la SOA. Las cuatro etapas de SGMM son:

- Planear y determinar el enfoque del Gobierno.
- Definir el modelo de Gobierno SOA.
- Permitir e implementar el modelo de Gobierno SOA.

²³ SADTLER, Carla. HUBER, Philipp. YI, SangMin. Enabling SOA Using WebSphere Messaging. IBM Redbooks. 2006. p. 65.

- Medir y refinar el modelo de Gobierno SOA.

Un producto para ayudar a llevar a cabo el ciclo de vida del Gobierno SOA en general y de SGMM, específicamente, es el complemento de Gobierno de SOA para el Compositor del Método Racional de IBM. Los servicios comerciales internacionales (GBS) de IBM pueden generar el accionar de un Centro de Servicios web/SOA de IBM de excelencia para ayudar a su organización a establecer un centro de excelencia de SOA y desarrollar prácticas de Gobierno.

2.2.1.6 Definición del servicio. El aspecto más fundamental del Gobierno SOA es supervisar la creación de servicios. Los servicios deben identificarse, su funcionalidad debe describirse, su comportamiento debe definirse y las interfaces deben diseñarse. El COE (Centro de Excelencia) del Gobierno podría no realizar estas tareas, pero se asegura de que se lleven a cabo. El COE coordina los equipos que crean y requieren servicios para asegurarse de que se satisfacen sus necesidades y se evita un esfuerzo duplicado.

A menudo, no es obvio lo que debería ser un servicio. La función debería coincidir con un conjunto de tareas comerciales repetitivas. Las limitaciones del servicio deberían encapsular una capacidad reutilizable libre de contexto. La interfaz debería exponer qué hace el servicio, pero ocultar cómo se implementa dicho servicio y permitir que cambie la implementación o implementaciones alternativas. Cuando los servicios se diseñan desde lo reutilizable, pueden diseñarse para modelar la empresa; cuando incluyen una función existente, puede ser más difícil crear e implementar una buena interfaz comercial.

Un ejemplo existente de las dificultades posibles en la definición de los límites del servicio es dónde establecer límites transaccionales. Generalmente, un servicio funciona en su propia transacción al asegurarse de que su funcionalidad funcione completamente o que se revoque en su totalidad. Sin embargo, un coordinador del

servicio (también conocido como, maestro compositor o coreógrafo) podría querer invocar servicios múltiples en una sola transacción (preferentemente mediante una interacción específica como las WS-AtomicTransactions). Esta tarea requiere que la interfaz del servicio exponga su soporte de transacción, entonces puede participar en la transacción del llamador. Pero dicha exposición requiere confianza en el llamador y puede ser riesgoso para el proveedor.

Por ejemplo, el proveedor podría bloquear los recursos para llevar a cabo el servicio, pero si el llamador nunca termina la transacción (falla en la confirmación o la revocación), el proveedor tendrá la dificultad de lanzar los bloqueos de recursos de manera correcta. Como lo muestra este escenario, el alcance de un servicio y quien tiene el control a veces no es una decisión fácil²⁴.

2.2.1.7 Control de versiones del servicio. Tan pronto como un servicio se encuentra disponible, los usuarios de esos servicios comienzan a necesitar cambios. Se necesita solucionar los errores lógicos, agregarse nuevas funcionalidades, rediseñar interfaces y eliminar la funcionalidad innecesaria. El servicio refleja la empresa, entonces a medida que cambia la empresa, el servicio necesita cambiar según corresponda.

Sin embargo, con los usuarios existentes del servicio, se necesita realizar los cambios con un buen criterio con el fin de no interrumpir su operación exitosa. Al mismo tiempo, las necesidades de los usuarios existentes no pueden permitirse impedir las necesidades o funcionalidades adicionales que los usuarios anhelan²⁵.

²⁴ Ibid., p. 96.

²⁵ SOA BLUEPRINT, op. cit.

El control de las versiones del servicio cumple estas metas contradictorias. Permite a los usuarios estar satisfechos con un servicio existente para continuar utilizándolo sin cambios y permite que evolucione el servicio para satisfacer las necesidades de los usuarios con nuevos requisitos. La interfaz del servicio actual y el comportamiento se preservan como una versión, mientras que el servicio más nuevo se presenta como otra versión. La compatibilidad de versiones puede permitir al consumidor esperar que una versión invoque una versión diferente, pero compatible.

Mientras que la versión ayuda a resolver estos problemas, también introduce otros nuevos, como la necesidad de migrar.

2.2.1.8 Migración del servicio. Incluso con el control de versiones del servicio, un consumidor no puede depender de un servicio (o, más específicamente, una versión deseada de ese servicio) para que esté disponible y que sea compatible para siempre, ya que finalmente, el proveedor de un servicio está obligado a dejar de proporcionarlo. La compatibilidad de la versión puede ayudar a retrasar la "hora de la verdad", pero no suprimirla. El control de las versiones no obsoleta el ciclo de vida del desarrollo del servicio, pero permite que el ciclo de vida afecte generaciones sucesivas.

Cuando un consumidor comienza a utilizar un servicio, está creando dependencia en ese servicio, una dependencia que debe ser manejada. Una técnica de gestión es la migración planificada y periódica para las versiones más nuevas del servicio. Este enfoque también permite al consumidor aprovechar la ventaja de funciones adicionales agregadas al servicio²⁶.

²⁶ DURVASULA, op. cit.

Sin embargo, incluso en empresas con el mejor Gobierno, los proveedores de servicios no pueden depender solo de la migración del consumidor. Por varias razones (código de legado, mano de obra, presupuesto, prioridades), algunos consumidores podrían migrar a tiempo. ¿Eso significa que el proveedor debe sustentar la versión del servicio para siempre? ¿Puede el proveedor simplemente deshabilitar la versión del servicio un día después de que todos hayan migrado?

“Ninguno de estos dos extremos es conveniente. Un buen compromiso es un programa de desaprobación y caducidad planificado para cada versión del servicio, como se describió en el ciclo de vida de implementación del servicio”²⁷.

2.2.1.9 Modelo de mensajes del servicio. En una invocación al servicio, el consumidor y el proveedor deben acordar los formatos del mensaje. Cuando equipos de desarrollo independiente diseñan las dos partes, pueden tener la dificultad de encontrar fácilmente un acuerdo sobre los formatos de mensajes comunes. “Si se multiplica eso por docenas de aplicaciones con un servicio típico y una aplicación típica que utilice docenas de servicios se podrá ver cómo los formatos de mensajes de negociación simple pueden convertirse en una tarea a tiempo completo”²⁸.

Un enfoque común para evitar el caos en el formato del mensaje consiste en utilizar un modelo de datos canónicos. Un modelo de datos canónicos es un conjunto de formatos de datos que es independiente de cualquier aplicación y lo comparten todas las aplicaciones. De esta forma, las aplicaciones no tienen que acordar en los formatos de los mensajes, simplemente pueden acordar utilizar formatos de datos canónicos existentes.

²⁷ MICROSOFT CORPORATION. 2012, op. cit.

²⁸ KERRIE, Holley. ALI, Arsanjani. 100 SOA Questions Asked and Answered. Boston Massachusetts: Prentice Hall. 2011. p. 71.

Un modelo de datos canónicos abarca el formato de los datos en el mensaje, entonces necesita un acuerdo en torno al resto del formato del mensaje (como por ejemplo, campos de encabezado, qué datos contiene la carga útil de los mensajes y cómo se disponen los datos), pero el modelo canónico de datos es sumamente útil para alcanzar el acuerdo²⁹.

Un panel central de Gobierno puede actuar como una parte neutral para desarrollar un modelo de datos canónicos. Como parte de la inspección de las aplicaciones y del diseño de servicios, también puede diseñar formatos comunes de datos para utilizarlos en las invocaciones del servicio.

2.2.1.10 Supervisión del servicio. ¿Si un proveedor del servicio deja de trabajar, cómo se sabrá? ¿Se debería esperar hasta que las aplicaciones que usan esos servicios dejen de funcionar y las personas que los usan empiecen a verse afectadas?

Una aplicación combinada, una que combina varios servicios, es tan confiable como los servicios de los que depende. Dado que varias aplicaciones combinadas pueden compartir un servicio, una sola falla en el servicio puede afectar muchas aplicaciones. Debe definirse el ANS la confiabilidad y el desempeño en el que pueden depender los consumidores, dado que los consumidores deben ser supervisados para garantizar que cumplen con el ANS definido.

Una cuestión relacionada es la determinación de un problema. ¿Por qué deja de funcionar una aplicación combinada? Podría ser que la parte principal de la aplicación, la interfaz de usuario con la que interactúan los usuarios, haya dejado de funcionar. Pero también puede pasar que la parte principal funcione bien, pero que algunos de los servicios que utiliza o algunos de los servicios que esos servicios utilizan no estén funcionando correctamente. Por lo tanto, es importante

²⁹ BARRAZA, op. cit.

supervisar no solo cómo cada aplicación está funcionando, sino también cómo funciona cada servicio (como una colección de proveedores) y los proveedores individuales. Es de suma importancia la correlación de eventos entre servicios en una sola transacción comercial.

Dicha supervisión puede ayudar a detectar y a evitar problemas antes de que sucedan. Puede detectar desequilibrios de carga y cortes, al proporcionar alertas antes de que la situación sea crítica, y también puede intentar corregir los problemas de forma automática. Puede medir el uso a través del tiempo para ayudar a predecir los servicios que están siendo cada vez más populares, entonces pueden funcionar con una capacidad en aumento.

2.2.1.11 Propiedad del servicio. Cuando varias aplicaciones combinadas utilizan un servicio, ¿quién es responsable de ese servicio? ¿Esa persona u organización es responsable de todas ellas? Una de ellas; si es así, ¿cuál? ¿Otros piensan que son propietarios del servicio? Es entonces aquí cuando nos enfrentamos al “mundo ambiguo de la propiedad del servicio”.

“Es difícil de adquirir cualquier recurso compartido y cuidarlo, ya sea un parque de un vecindario, un marco de trabajo Java reutilizable o un proveedor de servicio. Además, un recurso de agrupación necesitado proporciona un valor más allá de cualquier costo del participante, por ejemplo un sistema de caminos públicos”³⁰.

A menudo, una empresa organiza su la estructura informativa de su personal y las finanzas entorno a las operaciones comerciales. En la medida en que una SOA organiza la IT de la empresa entorno a las mismas operaciones, el departamento responsable por algunas operaciones también puede ser responsable del desarrollo y el tiempo de ejecución de la IT para esas operaciones. “El departamento posee esos servicios. Dado que los servicios y las aplicaciones

³⁰ SOA BLUEPRINT, op. cit.

combinadas en un SOA a menudo no siguen una estructura financiera e informativa estrictamente jerárquica, lo que genera brechas y solapamiento en las responsabilidades de IT”³¹.

Una cuestión relacionada en los roles de los usuarios es que sado que el enfoque de la SOA consiste en alinear la IT, la empresa y la reutilización de dicha empresa, muchas personas diferentes en una organización tienen voz y voto en lo que serán los servicios, cómo funcionarán y cómo los utilizarán. Estos roles incluyen analistas comerciales, arquitectos empresariales, arquitectos de software, desarrolladores de software y administradores del IT. Todos estos roles tienen un interés en asegurarse de que los servicios cumplan con las necesidades de la empresa y funcionen bien.

Una SOA debe reflejar su empresa. Generalmente, esto implica cambiar la SOA para ajustarse a la empresa, pero en algunos casos, podría ser necesario cambiar la empresa para coincidir con la SOA. Cuando esto no es posible, se necesitan niveles de cooperación en aumento entre varios departamentos para compartir la complicación de desarrollar servicios comunes. Esta cooperación puede alcanzarse mediante un comité vigente de organización cruzada que, en efecto, posea los servicios y los administre.

2.2.1.12 Pruebas del servicio. El ciclo de vida de implementación del servicio incluye la etapa de prueba, durante la cual el equipo confirma que un servicio funciona adecuadamente antes de activarlo. Si se prueba un proveedor del servicio y parece funcionar correctamente, ¿el consumidor necesita volver a probarlo? ¿Todos los proveedores de un servicio se prueban con el mismo rigor? Si un servicio cambia, ¿necesita volver a probarse?

³¹ SADTLER, op. cit, p. 205.

La SOA aumenta la oportunidad de probar la funcionalidad en aislamiento y aumenta la expectativa de que funciona como se debe. Sin embargo, la SOA también introduce la oportunidad de volver a probar la misma funcionalidad de forma repetida por cada nuevo consumidor que necesariamente no cree que los servicios que usa funcionen de forma adecuada y consistente. “Mientras tanto, dado que las aplicaciones combinadas comparten servicios, un solo servicio de errores lógicos puede afectar de forma adversa un rango de aplicaciones aparentemente no relacionadas de esos errores de programación”³².

Para apalancar los beneficios de la reutilización de la SOA, los consumidores del servicio y los proveedores necesitan acordar un nivel adecuado de evaluación de los proveedores y necesitan asegurarse de que la evaluación se realice como se acordó. Entonces, un consumidor de servicios solo necesita probar su propia funcionalidad y las conexiones con el servicio y puede asumir que el servicio funciona como se definió.

2.2.1.13 Seguridad del servicio. ¿Se le permitirá a alguien invocar un servicio? ¿Debería un servicio con un rango de usuarios permitir a todos los usuarios acceder a todos los datos? ¿Necesitan protegerse los datos intercambiados entre los consumidores del servicio y los proveedores? ¿Un servicio necesita ser tan seguro como las necesidades en sus usuarios más precavidos o como aquellos de sus usuarios más apáticos?

La seguridad es una proposición difícil, pero necesaria para cualquier aplicación. La funcionalidad necesita limitarse a los usuarios autorizados y los datos necesitan protegerse desde la interceptación. Al proporcionar más puntos de acceso para la funcionalidad (es decir, servicios), la SOA tiene el potencial de aumentar en gran medida la vulnerabilidad en las aplicaciones combinadas.

³² BASTIDA, Leire. Gobierno SOA: Elemento Clave en la Integración de Negocio y Tecnología. Sevilla: JSWEBOCT08. 2008. p. 8.

La SOA crea servicios que son fáciles de reutilizar, incluso por los consumidores que no ayudarán a volver a utilizarlos. Incluso entre los usuarios autorizados, no todos los usuarios deberían tener acceso a todos los datos a los que tiene que acceder el servicio. Por ejemplo, un servicio para acceder a las cuentas bancarias debería solo hacer que las cuentas de usuarios particulares se encuentren disponibles, aunque el código también tenga acceso a otras cuentas para otros usuarios. “Algunos consumidores de un servicio tienen más necesidades que otros consumidores del mismo servicio para la confidencialidad de los datos, la integridad y el no rechazo”³³.

Las tecnologías de invocación de servicio debe ser capaz de proporcionar todas esas capacidades de seguridad. El acceso a los servicios tiene que controlarse y limitarse a los consumidores autorizados. La identidad del usuario debe propagarse en servicios y utilizarse para autorizar el acceso a datos. Las calidades de la protección de los datos deben ser representadas como políticas dentro de rangos. Esto permite a los consumidores expresar niveles mínimos de protección y capacidades máximas y de unirse con proveedores adecuados que podrían, de hecho, incluir protecciones adicionales.

2.2.2 ORACLE

2.2.2.1 Mejores Prácticas SOA, Estándares y Gobernabilidad. En esta era, las aplicaciones empresariales han comenzado la transición de las aplicaciones guiadas por las interfaces de usuarios a “arreglos” de servicios interoperables que de hecho son también reusables. Estos servicios representan funciones de negocio cuyo fin es la de poder ser ensamblados de forma tal que representen una nueva aplicación.

³³ DEEPAL, Jayasinghe. Open Source Competition - Mule Galaxy vs WSO2 Registry. WSO2 Portal. 2008. p. 51.

Una de las mayores ventajas de tal cambio en la arquitectura de las aplicaciones es que el servicio puede ser ahora reutilizado en la evolución y desarrollo de un proceso de negocio. Los usuarios necesitan entender que tal acercamiento a la construcción de aplicaciones compuestas y procesos de negocios no funcionara en la ausencia de una plataforma de cumplimiento de estándares para la construcción de tales servicios.

La interoperabilidad puede ser un reto debido al hecho de que los protocolos de servicios de Web para tales actividades como mensajería, optimización y confiabilidad pueden ser un tanto complicados. Los servicios pueden, de hecho, residir a través de diferentes plataformas. Sin una plataforma que haya sido diseñada alrededor de estándares y con la intención de lograr interoperabilidad, pudiera no ser posible armar rápidamente los servicios con la intención de lograr cumplir con los requerimientos del negocio.

Un tipo de “middleware” desarrollado por Oracle, Oracle Fusion Middleware, ha sido construido basado en una infraestructura común. Ha sido diseñado para utilizar estándares de industria para todas las capacidades de “Service Oriented Architecture” (SOA). Protocolos de calidad de servicio y optimización de mensajes son provistos como un medio de mejorar funcionalidad. Estas herramientas pueden ser utilizadas fuera de la lógica del negocio al igual que la estrategia de implementación de los servicios.

El negocio también se beneficia de los estándares que describen no sólo como las aplicaciones cooperan, sino también como son estas construidas. A través de la utilización de estándares de SOA, los negocios pueden escapar de los obstáculos de las plataformas, mientras los desarrolladores podrán ser más rápidos con nuevas destrezas fácilmente transferibles.

En corto tiempo, la próxima generación de estándares de desarrollo estará enfocada en dos áreas principales, el proveer un modelo común para el control de la relación entre los servicios; y permitir a los desarrolladores implementar servicios web robustos utilizando objetos Java comúnmente utilizados.

2.2.2.2 Estándares. Aun cuando ha habido una explosión de propuestas de servicios web, la plataforma estándar de SOA está construida en una plataforma basada en las siguientes tres áreas: implementación; mensajería; y descripción y descubrimiento de servicios.

2.2.2.3 Mensajería. De manera que los servicios puedan comunicarse entre ellos, los mensajes deben ser codificados de acuerdo a las especificaciones SOAP 1.1 y SOAP 1.2 (SOAP es el acrónimo para “Simple Object Access Protocol”, para el intercambio de información estructurada en la implementación de servicios de WEB). Los mensajes son luego intercambiados a través de HTTP. Los estándares SOAP conforman la fundación de interoperabilidad.

Dos estándares vitales que han sido utilizados como el mecanismo para ayudar en las capacidades de compresión y eficiencia dentro de los mensajes SOAP que incluyen contenido binario son MTOM (Message Transmission Optimization Mechanism) y SOAP con agregados.

La fábrica de servicios de Oracle hace uso de ambos optimizadores como un medio para asegurar que los servicios están disponibles para comunicar de la manera más eficiente posible. Como SOAP provee sólo lo básico en el intercambio de mensajes, mucha más data es necesaria para proveer las directrices en la eventualidad de intercambio asíncronico. “WS-Addressing” define la cabecera de los mensajes los cuales son luego aplicados al mensaje SOAP de

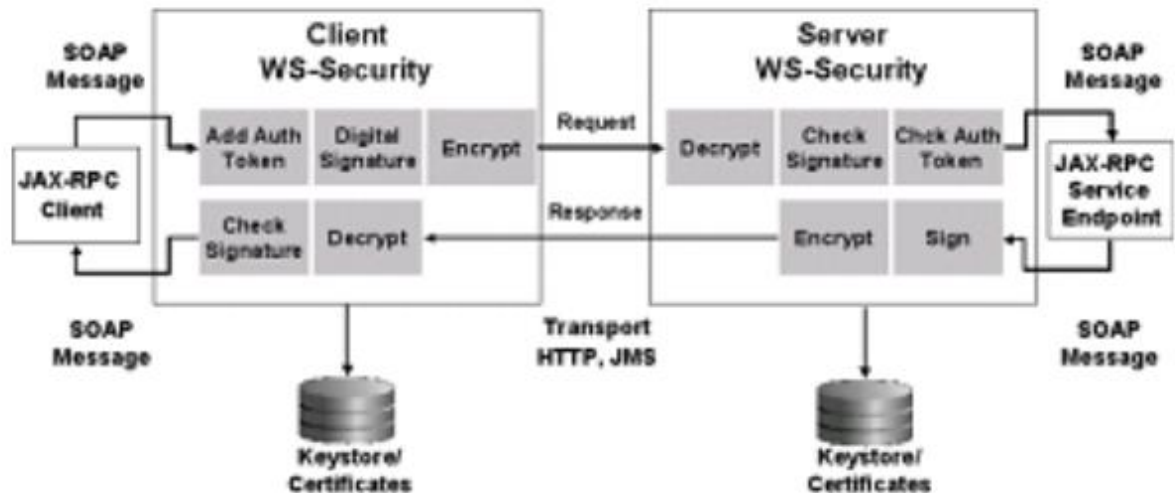
forma que se pueda determinar que respuestas pueden mejor ser enviados mientras que simultáneamente presentan la correlación entre los mensajes”³⁴.

Es vital proveer ciertas garantías de que los mensajes asincrónicos serán entregados al servicio que los ha invocado. Oracle ha trabajado junto a los emisores de estándares OASIS en ser el co-autor y en dirigir la convergencia de la industria alrededor del protocolo de “WS-Reliable Messaging” o WS-RM el cual es utilizado por los servicios como mecanismo que provee las garantías alrededor de la entrega de los mensajes y el orden en el que los mensajes son procesados.

Otro importante componente de los servicios web sobre todo para las aplicaciones empresariales es Seguridad. Oracle es una de las compañías que ha liderado los trabajos en muchos de los comités de estándares alrededor de la seguridad en adición a la federación de la identidad vía la utilización de los servicios web. Los mensajes SOAP son particularmente protegidos vía los estándares de “WS-Security”, los cuales son ahora utilizados como medio en la definición de la manera en que la encriptación, autenticación y firmas digitales serán utilizados como mecanismos para asegurar las comunicaciones.

³⁴ MULE GALAXY. Mule Galaxy: Mule Source's Open Source SOA [Online]. 2014. Disponible en: <www.mulesource.org/display/GALAXY/Home>

Figura 3. Resumen de la arquitectura WS-Security en OracleAS



Fuente: AFSHAR, M³⁵.

2.2.2.4 Metadata para la Descripción y Descubrimiento de los Servicios.

Oracle's Service Fabric utiliza metadata estándar para describir los mensajes y protocolos usados por los servicios web. Estas metadata estándar son utilizadas por las aplicaciones y las infraestructuras para garantizar que los servicios puedan interoperar basado en los requerimientos de los servicios colocados por los usuarios. Los más importantes estándares de metada son WSDL, WS_Policy, WS_MetadaExchange y UDDI.

WSDL, describe el mensaje que un servicio puede recibir y enviar. Es el más básico lenguaje de "contrato" utilizado para describir la funcionalidad ofrecida por un servicio.

WS_Policy, describe la calidad de las características del servicio y los requerimientos asociados a este servicio. Típicamente las políticas

³⁵ AFSHAR. 2007, op. cit, p. 13.

describen los requerimientos de seguridad de un servicio, la optimización soportada por un servicio tal como MTOM y si un servicio utiliza WS-ReliableMessaging. Oracle Web Services Policy Manager provee las herramientas para construir y apoyar las políticas para Oracle Fusion Middleware³⁶.

WS-MetadataExchange es un protocolo de “handshake” que permite al usuario el obtener documentos WSDL y WS_Policy asociados a un servicio.

UDDI es un modelo utilizado para el registro de los servicios web. Provee un repositorio común de metadata relacionado a los servicios que pueden ser utilizados para descubrir que servicios están disponibles y para seleccionar los servicios que están disponibles para ser utilizados para la construcción de nuevos servicios “compuestos” y procesos de negocio. Oracle Service Registry representa la implementación de UDDI bajo Oracle Fusion Middleware.

2.2.2.5 Servicios de Implementación. JAX-RPX y JAX-WS son los estándares definidos por la comunidad de Java que describe como los desarrolladores de Java pueden crear servicios Web. JAXRPC es el estándar que es utilizado en J2EE 1.4; JAX-WS, como muchos otros de las nuevas especificaciones de JavaEE eliminan mucha de la complejidad asociada con el desarrollo de servicios Web. Ejemplo de esto es que el JAX-WS provee un modelo simple para la implementación de la lógica del negocio y exponen el “contrato” como una interfaz WSDL utilizando anotaciones en el código implementado. Al apalancarse en JAX_WS, los desarrolladores construyen servicios portables utilizando habilidades que pueden ser ampliamente aplicados a través de muchos productos.

³⁶ Ibid., p. 11.

Figura 4. Annotated Java Class for Web Services

```
import javax.jws.WebMethod;  
import javax.jws.WebService;  
  
@WebService  
public class Echo {  
  
    @WebMethod  
    public String echoString(String p) {  
        return "echo" + p;  
    }  
}
```

Fuente: AFSHAR, M³⁷.

2.2.2.6 Mejores prácticas SOA. SOA se contempla como el componente clave para mejorar la eficiencia de la tecnología de información. Pero para poder implementar SOA en un negocio, se requiere de tener mucho más que manejo de la tecnología. Es esencial tener habilidades en las prácticas gerenciales.

Las implementaciones de Service Oriented Architecture y las arquitecturas empresariales se definen como independientes de las implementaciones de tecnología. “Con este drástico cambio en el negocio, es importante buscar nuevas maneras de estructurar los negocios mientras se mantiene la integridad de IT. La clave para lograr esto es a través de la identificación de maneras de definir y desarrollar arquitecturas tal como SOA”³⁸.

2.2.2.7 Gobernabilidad. Debido a los numerosos “issues” y retos que existen dentro de la industria de la computación, el rol de IT ha sido drásticamente alterado. En esta era, IT tiene que reaccionar de manera rápida y eficiente de

³⁷ Ibid., p. 9.

³⁸ DEEPAL, op. cit, p.126.

manera que permita a las transacciones de negocio ocurrir en tiempo real. IT ha sido forzado a manejar y diseñar parte de una compleja y altamente integrada arquitectura empresarial. Lo que esto significa es que los límites entre el lado del negocio de una operación y el lado de IT han quedado borrosos.

Gobernabilidad provee una estructura que apoya los objetivos del negocio de clientes a un número de niveles, incluyendo estrategia, funcionamiento, y operaciones. Gobernabilidad provee las reglas, las métricas, los procesos, al igual que las estructuras organizacionales que son necesarias para una planificación efectiva, dirección, toma de decisiones y control de los compromisos de SOA como un mecanismo de lograr las necesidades de negocio de los clientes al igual que los retos de sus metas. “Un modelo de gobernabilidad SOA deberá definir que tiene que ser hecho, como hacer para cumplir, al igual que definir como deberá ser medido efectivamente”³⁹.

A continuación alguna de las más importantes preguntas que ayudan en la definición de una buena estructura de gobernabilidad en los compromisos SOA:

- ¿Cuáles son las ventajas de los clientes en tal compromiso? ¿Cuáles son las expectativas y objetivos del cliente?
- ¿Cuáles son los roles, procedimientos, estructuras y responsabilidades que ya existen en el sitio del cliente para la planificación, toma de decisiones y dirección de IT?
- ¿Cómo pueden ser desarrolladas las habilidades y el liderazgo?
- ¿Cuáles son las guías y principios necesarios para la optimización del alineamiento de IT y el negocio?

³⁹ MULE GALAXY, op. cit.

- ¿Cuál sería una buena forma de estructurar el negocio y a IT de manera que ambos puedan interactuar manteniendo la consistencia a la vez que se mantenga la flexibilidad suficiente para poderse adaptar a los cambios de los servicios que actualmente ocurren?
- ¿Qué niveles de servicio, descripción y estandarización de definición de servicios es apropiado?
- ¿Cómo los servicios y los proveedores de servicios pueden ser mejor medidos y controlados? ¿Quién debe monitorearlos, autorizarlos y definir los cambios a los servicios que existen?
- ¿Cómo se definirán las estrategias de identificar las fuentes de los servicios?
- ¿Cuáles son los actuales problemas? ¿Cómo será el cliente ayudado a resolverlos?

Parece ser que un modelo de gobernabilidad formalizado y aceptado es la clave para el éxito de los objetivos del negocio. Entonces, es recomendable el establecimiento de las funciones de gobernabilidad en los compromisos SOA.

El modelo de gobernabilidad de manera simultánea, deberá lograr el requerimiento fundamental de adaptación incremental, mientras se enfoca en utilizar las lecciones aprendidas en cada paso como un medio de definir y ejecutar el próximo paso. De manera que el modelo de gobernabilidad sea un éxito, un nuevo cuerpo de gobernabilidad debe ser creado para la implementación y adaptación de SOA. Para lograr un alto grado de aceptación, es recomendable utilizar la organización existente, trabajando junto a ellos para adaptarlos a los compromisos de SOA

2.2.2.8 Principios de Gobernabilidad. La definición de una estrategia de dirección del cliente es necesaria para lograr el éxito en el desarrollo de una apropiada implementación SOA, mientras de manera simultánea se mantiene foco sobre las necesidades del negocio. “Un firme entendimiento de ambos, los objetivos del negocio y la estrategia, son necesarios tanto para IT como para el lado operativo del negocio. Los principios de gobernabilidad y las guías, son necesarias para formar las bases primarias de cualquier decisión mayor”⁴⁰. Tales principios son necesarios para darle forma a las áreas de solución, mientras se definen las maneras en el cual una colaboración efectiva ocurre. Entonces, los principios de gobernabilidad deberán ser bien entendidos por aquellos que laboran al nivel gerencial ejecutivo de la empresa. Los dos principales son:

- **Gobernabilidad Centralizada.** Esta es mejor para la empresa. El consejo de gobernabilidad está representado por miembros del nivel de servicio. Se encuentran también expertos sobre algunas materias los cuales son capaces de comunicarse con las personas responsables de la implementación de componentes claves de tecnología de la solución. Este consejo revisará cualquier adición o eliminación a la lista de servicios, al tiempo que revisaran los cambios a servicios existentes y autorizaran cualquier cambio de implementación necesario.
- **Gobernabilidad Distribuida.** Este modelo trabaja mejor en los equipos distribuidos. Cada unidad de negocio tendrá el control de cómo se proveerán los servicios dentro de su propia organización. Entonces, un acercamiento de servicio funcional (por área funcional del negocio) es necesario. Guías y estándares principales son provistos a los diferentes equipos por el comité central.

Los principios básicos (o principales) para el diseño de la arquitectura y de la definición de los servicios pueden ser extraídos de estas guías principales las

⁴⁰ AMBERPOINT. Actional Solutions for SOA Governance [Online]. 2015. Disponible en: <<http://www.actional.com/solutions/soa-governance/>>

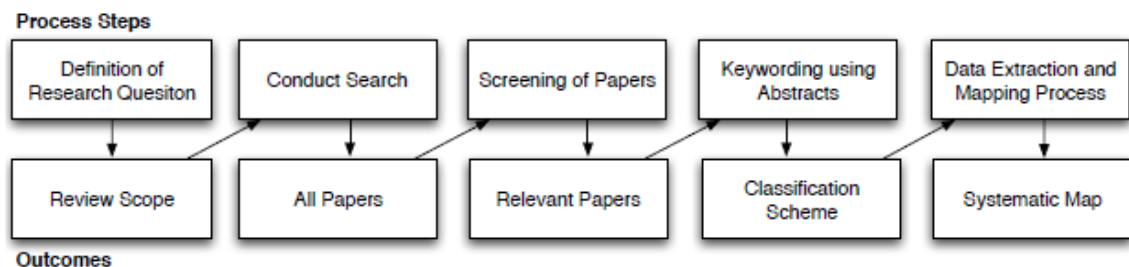
cuales están enfocadas en temas particulares. Tales principios caracterizan el comportamiento intrínseco para el tipo de implementación, mientras cubre las guías principales y específicas de los principios de arquitectura del proyecto.

A través de la posesión de un entendimiento completo de los principios de SOA, una decisión más sabia puede ser tomada con relación a la aplicabilidad de SOA a los problemas de diseño. Tales principios dirigen muchas de las características que son necesarias al diseñar un servicio. Cada una de las características debe ser encontrada en uno o más de los principios de diseño SOA el cual provee integridad a los principios y a las características por igual.

3. METODOLOGÍA. ESTUDIO DE MAPEO SISTEMÁTICO

Petersen et al⁴¹ define un estudio de mapeo sistemático como un estudio secundario que tiene como objetivo construir un esquema de clasificación y estructurar un campo de interés de la ingeniería de software. Para llevar a cabo el estudio sugiere un proceso de cinco pasos que son los siguientes: definir las preguntas de investigación, realizar la búsqueda de los documentos pertinentes, seleccionar los estudios primarios, análisis de los resúmenes y extracción de palabras clave, extracción de datos, y por último mapear los estudios primarios seleccionados. Cada uno de los pasos del proceso tiene un resultado explicado en cada uno de los numerales del artículo, siendo el mapeo sistemático el resultado final de los procesos. Para llevar a cabo el presente estudio se siguen los pasos mencionados anteriormente, y se construye un protocolo con el propósito de evitar sesgos en el estudio⁴². En el siguiente diagrama se representa dicho proceso:

Figura 5. Estudio de mapeo sistemático



Fuente: MOORE, G. TEGAN, W⁴³.

⁴¹ PETERSEN, op. cit, p. 2.

⁴² ELBERZHAGER, F. MÜNCH, J. NHA, V. T. N. A systematic mapping study on the combination of static and dynamic quality assurance techniques. Information and Software Technology. 2011. p. 6.

⁴³ MOORE, G. TEGAN, W. Systematic Mapping Studies In Software Engineering. Boston: IEEE Xplore. 2005. p. 11.

3.1 PREGUNTAS DE INVESTIGACIÓN

Las preguntas de investigación se definen de acuerdo a los objetivos principales del estudio que son brindar una serie de estrategias para la conformación de un Gobierno y su portafolio de servicios en una implementación SOA que puedan ser implementadas por cualquier arquitecto o ingeniero de software para que alcancen el cumplimiento de sus objetivos. Así como también determinar los criterios de selección más importantes para los arquitectos e ingenieros de software al momento de decidirse por una herramienta de gobierno que permita la gestión adecuada del portafolio de servicios teniendo en cuenta las necesidades específicas de negocio que requiera su implementación SOA. Entonces, para poder lograr los objetivos, se definieron dos preguntas de investigación (PI).

La pregunta PI1, define la base del estudio y brinda una perspectiva general de las estrategias para la conformación de un Gobierno SOA y su portafolio de servicios. Se define de la siguiente forma: ¿Qué estrategias para la conformación de un Gobierno y su portafolio de servicios en una implementación SOA se deben brindar a los arquitectos software para que alcancen el cumplimiento de sus objetivos?

La pregunta PI2 determina los criterios de selección de una herramienta de gobierno que permita la gestión adecuada del portafolio de servicios. Se define de la siguiente forma: ¿Cuáles son los criterios de selección más importantes para los arquitectos e ingenieros de software al momento de decidirse por una herramienta de gobierno que permita la gestión adecuada del portafolio de servicios?

3.2 SELECCIÓN DE BIBLIOTECAS DIGITALES Y DEFINICIÓN DE LA ESTRATEGIA DE BÚSQUEDA

Se consideraron las siguientes bibliotecas digitales para realizar la búsqueda: Scopus e IEEEExplore. La razón para seleccionarlás es que según Dieste et al⁴⁴, Scopus tiene menos debilidades que otras bases, cubre una amplia gama de publicaciones en el campo de la ciencia de la computación y mantiene una base de datos completa y consistente. Además de Scopus, algunos autores recomiendan el uso de otras bibliotecas digitales como IEEEExplore, Inspec, Compendex y ACM Digital Library⁴⁵. De las últimas posibles se selecciona IEEEExplore.

Los términos utilizados en las búsquedas se formaron mediante la combinación de términos derivados del tema de interés y de los términos relacionados con experimentos. Los términos del tema de interés fueron obtenidos de algunas sugerencias de varios autores y se fueron ajustando durante las búsquedas preliminares realizadas. Mediante el uso del signo *, en las bibliotecas digitales quedan incluidos en la búsqueda los sinónimos de dicho término. En la TABLA 1 se presentan los términos usados en la cadena de búsqueda con los respectivos sinónimos considerados.

⁴⁴ DIESTE, O. GRIMÁN, A. JURISTO, N. Developing search strategies for detecting relevant experiments. Empirical Software Engineering. Vol. 14. 2009. p. 1.

⁴⁵ ELBERZHAGER, op. cit, p. 3.

Tabla 1. Términos y sinónimos usados para componer la cadena de búsqueda

Términos	Sinónimos
<i>Government</i>	<i>Governance</i>
	<i>Governability</i>
<i>Strategies</i>	<i>Model</i>
	<i>Patterns</i>
	<i>Framework</i>
<i>Portfolio</i>	<i>Catalog</i>
	<i>Registry</i>
	<i>Repository</i>
<i>Service-Oriented Architecture</i>	<i>SOA</i>

Fuente: Autor.

La cadena de búsqueda empleada se conformó con los términos de la TABLA 1 y con operadores booleanos. La cadena resultante es: (government OR governance OR governability) AND (strategies OR model OR patterns OR Framework) AND (portfolio OR catalog OR registry OR repository) AND SOA* AND service. En cada una de las bibliotecas digitales seleccionadas las búsquedas se realizaron considerando los campos palabras clave, título, resumen y artículos publicados entre los años 2006 y el 2015. La búsqueda en las bibliotecas digitales fue realizada del 11 de agosto de 2015 al 25 de agosto de 2015.

El motivo de acotar la búsqueda al período 2006-2015, es que permite extraer una muestra amplia de artículos recientemente publicados y permite comparar algunos resultados con los artículos más similares (aunque el foco de los estudios sea diferente) y recientes al presente estudio.

3.3 SELECCIÓN DE ESTUDIOS

Para seleccionar los estudios primarios se definieron los siguientes criterios de inclusión y exclusión. Los criterios de inclusión definidos son los siguientes:

- Artículos de investigación sobre arquitecturas orientadas a servicios que reporten enfoques en Gobierno SOA y que aporten estrategias, tácticas, sugerencias, recomendaciones o mejores prácticas en la construcción, mantenimiento y evolución del portafolio o catálogo de servicios.

Los criterios de exclusión definidos son los siguientes:

- Desarrollos de software como Frameworks orientados únicamente a los aspectos técnicos del Gobierno SOA.
- Investigaciones sobre temas de diseño de SOA o de patrones de diseño que no contemplen el Gobierno dentro de su desarrollo.
- Estudios cuyo objetivo sea principalmente la evaluación de una herramienta y no la evaluación de una técnica.
- Artículos no disponibles para ser descargados.

El proceso de selección de estudios primarios comprende cuatro fases. De la búsqueda en las bibliotecas digitales resultaron 1483 artículos; 1140 obtenidos en Scopus y 343 obtenidos en IEEEExplore.

La primera fase de la selección consiste en excluir los artículos duplicados. De 1483 artículos, se quitaron 142 artículos duplicados. Los 1341 artículos resultantes de la primera fase, se tomaron como entrada de la segunda fase, en la que se

aplicaron los criterios de inclusión y exclusión basados en el título de los artículos. En la segunda fase se excluyeron 1199, por lo que quedaron 142 artículos. Los artículos excluidos en la segunda fase principalmente pertenecen a otro dominio, como por ejemplo artículos relacionados con gestión de proyectos y seguridad informática.

En la fase 3, se aplican los criterios de inclusión y exclusión basados en el resumen de los artículos primarios. De un total de 142 artículos se excluyen 62. La razón principal de la exclusión en esta fase es que algunos artículos no reportan estudios relacionados con Gobierno SOA y mucho menos con conformación del portafolio de servicios.

En la última fase, la fase 4, se aplicaron los criterios de inclusión y exclusión basados en el texto completo de los artículos primarios. A 80 artículos resultantes de la fase 3 se les aplicó los siguientes criterios de calidad tomados de Elberzhager et al⁴⁶, para comprobar su relación a las preguntas planteadas en el presente estudio:

- Los objetivos de la investigación se describen con claridad.
- El enfoque se explica suficientemente.
- Los factores contextuales y ambientales se presentan con claridad.
- Los datos de entrada y salida de usar el enfoque son explícitamente mencionados.
- El tema del enfoque se aclara suficientemente.

⁴⁶ Ibíd., p. 13.

- La evidencia del enfoque se documenta.

Aplicando los criterios de calidad mencionados, de 80 artículos, en la fase 4, se descartaron 32 artículos, resultando 48 artículos primarios que se toman como entrada para el proceso de extracción de datos.

3.4 EXTRACCIÓN DE DATOS Y SÍNTESIS

Para la extracción de datos se desarrolló un formulario con el fin de extraer toda la información relevante para contestar las preguntas de investigación. Además se creó un esquema de clasificación para los artículos. El esquema se compone de la siguiente forma:

- Tipo de técnica. se refiere al tipo de técnica usada para validar la estrategia de conformación de gobierno y/o portafolio de servicios. Las técnicas encontradas en los experimentos en gran medida corresponden a casos de estudio y a problemas comunes estandarizados.
- Aspectos estudiados. se refiere a los principales aspectos estudiados y conceptos que evidencien una contribución al artículo de investigación. Los aspectos y conceptos encontrados en los objetivos de los artículos primarios se agruparon en las categorías que se describen en la TABLA 2.
- Contexto. se refiere al tipo de programa empleado para realizar los estudios. Permite diferenciar estudios que se realizaron con programas de pequeña escala o programas de escala industrial.

Tabla 2. Categorías de clasificación de los estudios

Categoría	Descripción	Cant. de Artículos
Estrategias para la conformación de Gobierno en una SOA	Agrupar todas las estrategias, tácticas, recomendaciones, sugerencias o buenas prácticas para diseñar y construir un Gobierno SOA.	33
Criterios para seleccionar una herramienta técnica de Gobierno.	Agrupar todos los criterios para seleccionar una herramienta de Gobierno que permita la gestión adecuada del portafolio de servicios y que brinde funcionalidades de valor en la implementación.	15

Fuente: Autor.

Los 48 artículos primarios obtenidos en el proceso de selección de estudios, se clasificaron según el esquema presentado para abordar las 2 preguntas planteadas en la presente investigación. La asignación de un artículo a una determinada categoría se realizó considerando los problemas en los que se focalizan los estudios o las técnicas que plantean para la conformación adecuada de un Gobierno SOA y su portafolio de servicios. Cada una de las categorías no son excluyentes, es decir que un artículo se puede haber clasificado en una o más categorías, dado que cada uno de los estudios tiene uno o más objetivos.

4. RESULTADOS

Los resultados del estudio de mapeo se presentan respondiendo cada una de las preguntas de investigación. Cada una de las categorías de la TABLA 2 representa uno o un conjunto de temas de investigación y permiten responder la pregunta de investigación PI1. La pregunta de investigación PI2 se responde con los artículos que se clasifican en la categoría llamada “Criterios para seleccionar una herramienta técnica de Gobierno”.

4.1 PREGUNTA DE INVESTIGACIÓN 1

¿Qué estrategias para la conformación de un Gobierno y su portafolio de servicios en una implementación SOA se deben brindar a los arquitectos software para que alcancen el cumplimiento de sus objetivos?

4.1.1 Selección del personal. Es necesario contar con un equipo de trabajo que se encargué directamente del Gobierno SOA. Para esto es preciso conocer cuál será el alcance de dicha estrategia, generando así un listado de exigencias que deberán cumplir las personas seleccionadas⁴⁷. Este listado se traducirá en competencias que deberán tener para ser parte del equipo, seleccionando así a los candidatos más adecuados⁴⁸.

4.1.2 Conformación del equipo de trabajo. Conformar el equipo de trabajo no es una tarea fácil. Es importante que el equipo tenga actitud, conocimiento, carácter y

⁴⁷ ABRAN, A. MOORE, J. W. Guide to the software engineering body of knowledge (swebok). IEEE Press. 2004. p. 6.

⁴⁸ TRACEY, N. CLARK, J. MANDER, K. Automated program flaw finding using simulated annealing. ACM SIGSOFT Software Engineering Notes. Vol. 23. 2008. p. 8.

empatía entre sus participantes⁴⁹. Para conformar el equipo se recomienda tener en cuenta los siguientes aspectos:

- Visión común. Definir un rumbo de trabajo ofrece la posibilidad a cada miembro del equipo de interiorizar el camino que debe recorrer, aportando soluciones creativas a los problemas que puedan presentarse. Esta visión debe ser discutida por el equipo creando un compromiso de trabajo, lo que mejorará el entendimiento al punto de llegada⁵⁰. Sin embargo sí esta visión no es conocida y comprendida, el equipo de trabajo perderá el rumbo y el entendimiento de las acciones que se encuentra realizando, produciendo una sensación de desconcierto y abandono⁵¹.
- Objetivos. Los objetivos deben ser reales y alcanzables. Es preferible alcanzar varios objetivos pequeños que nunca alcanzar un solo objetivo gigantesco. El equipo se medirá implícitamente por su capacidad para lograr sus objetivos, incumplirlos podría frustrar el espíritu del mismo⁵².
- Tamaño y límites. Es necesario fijar el tamaño y límites del equipo, esto permitirá tener un marco de referencia y operación definido, a su vez favorecerá el establecimiento de procedimientos con un alcance medible⁵³.
- Roles. La identificación de los roles dependerá del método utilizado para la implantación del Gobierno SOA, y estará directamente relacionado con el equipo de trabajo, así como con la estructura organizacional⁵⁴.

⁴⁹ LAKHOTIA, K. MCMINN, P. HARMAN, M. An empirical investigation into branch coverage for C programs using CUTE and AUSTIN. Journal of Systems and Software. Vol. 83. 2010. p. 3.

⁵⁰ MANSOURI, S. A. ZHANG, Y. Search based software engineering: A comprehensive analysis and review of trends techniques and applications. Department of Computer Science, King's College London, Tech. 2009. p. 5.

⁵¹ BALUDA, M. BRAIONE, P. DENARO, G. PEZZÈ, M. Enhancing structural software coverage by incrementally computing branch executability. Software Quality Journal. Vol. 19. 2011. p. 8.

⁵² MOSCATO, P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program. Vol. 826. 2009. p. 7.

⁵³ WOHLIN. 2000, op. cit, p. 10.

- Liderazgo. La sana coordinación de la visión, así como el logro porcentual de cada objetivo está relacionado con una clara dirección, por ello, resulta relevante seleccionar a la persona que deberá cumplir este rol, sin olvidar la misión de generar un ambiente de empatía, para que cada miembro del equipo pueda aportar lo mejor de sí alineando el trabajo colectivo con la visión⁵⁵.

4.1.3 Capacitar al equipo de trabajo. La calidad de los servicios que ofrece una organización está directamente relacionada con la calidad en las capacitaciones que reciben sus miembros. El equipo de Gobierno SOA provee un conjunto de servicios, por lo tanto no capacitarlo sería pretender obtener un producto de alta calidad utilizando herramientas deficientes⁵⁶.

4.1.4 Seleccionar un conjunto limitado de Principios SOA. Existen varias propuestas de Principios SOA los cuales pueden servir de referencia, sin embargo es necesario verificar primero el grado de madurez de la Arquitectura Orientada a Servicios que exista en la Organización⁵⁷. Cada uno de estos principios, plantea la necesidad de desarrollar un conjunto de políticas y estándares, por lo tanto es recomendable realizar una selección pequeña de los mismos, con el fin de contar con los suficientes recursos (conocimientos, tiempo, personal, etcétera) para su desarrollo y evolución. Seleccionar una gran cantidad puede acarrear problemas operativos, al no contar con el personal necesario e idóneo para iniciar dicha labor⁵⁸.

4.1.5 Establecer un conjunto de políticas. Es importante fijar un rumbo y realizar iteraciones constantes, para medir la evolución de las políticas definidas.

⁵⁴ PETERSEN, op. cit, p. 11.

⁵⁵ ELBERZHAGER, op. cit, p. 14.

⁵⁶ DIESTE, op. cit, p. 5.

⁵⁷ IEEE. Recommended Practice for Architectural Description for Software-Intensive Systems, IEEE Standard 1471-2000, 2000. p. 10.

⁵⁸ WOHLIN, C. Experimentation in software engineering: Service-Oriented Architecture. Springer. Vol. 1. 2006. p. 2.

Recordando que uno de los propósitos de las políticas es dar una estructura unificada a los planes, evitando procesos repetitivos y forjando la delegación de funciones en pro del control. Las políticas desarrolladas deben ofrecer las directrices que guíen las acciones de los equipos de trabajo, sin la necesidad de consultar permanentemente a los niveles superiores, para determinar si se está obteniendo los resultados deseados. Por lo tanto cada política deberá facilitar las labores de control y la verificación de los resultados para la autorregulación y mejoramiento continuo a nivel de procesos⁵⁹.

Como punto de partida es posible definir un conjunto pequeño de políticas que sean aplicadas únicamente a un proyecto específico, o departamento, o unidad funcional. Esto permitirá medir los beneficios de dichas políticas y permitirá extender el uso a otras Unidades Funcionales con el fin de no generar traumas colectivos por pretender abarcar toda la Organización en una sola iniciativa global⁶⁰.

4.1.6 Plan de Comunicación. La planificación busca organizar las acciones que se realizarán para alcanzar un objetivo. El desarrollo de un plan de comunicación es una implementación de la planificación, el cual debe tener como objetivo principal comunicar la información necesaria al público seleccionado en el espacio y tiempo pertinente⁶¹. Es importante tener en cuenta los siguientes aspectos en el desarrollo del plan de comunicación:^{62 63 64}

- ¿Quién es el público?

⁵⁹ ERL. 2005, op. cit, p. 154.

⁶⁰ ZACHMAN, John. A framework for Information Systems Architecture. The IBM Systems Journal vol. 26. 1987. p. 5.

⁶¹ NIEMANN, M. CHRISTIAN, J. NICOLAS, R. RALF, S. Challenges of Governance Approaches for Service-Oriented Architectures. IEEE. 2009. p. 12.

⁶² AYATOLLAHZADEH SHIRAZI, M. R. Developing a More Comprehensive and Expressive SOA Governance Framework. China: 2nd IEEE International Conference on Information Management and Engineering. 2011. p. 5.

⁶³ AFSHAR. 2007, op. cit, p. 2.

⁶⁴ BROWN, W. SOA Governance - IBM's Approach. IBM Express. 2006. p. 7.

- ¿Cuál es el mensaje o mensajes?
- ¿Qué canales de comunicación se utilizarán?
- ¿Cómo será transmitido el mensaje?
- ¿Cuál es el propósito de la comunicación?

4.1.7 Plan de Capacitación. Este plan debe estar dirigido a mejorar e incrementar las habilidades del equipo de trabajo sin olvidar el alcance del Gobierno SOA, el cual puede incluir equipos de trabajo de otras áreas organizacionales⁶⁵. El éxito en el desarrollo de un Plan de Capacitación está relacionado con:

- ¿Cuál será el grado de conocimiento adquirido para el personal seleccionado después de recibir la capacitación con respecto a los objetivos que se persiguen?⁶⁶
- ¿Cómo se aplicarán las habilidades aprendidas en la capacitación con respecto a los objetivos que se persiguen?⁶⁷

Estos interrogantes ofrecen la posibilidad de eliminar fácilmente capacitaciones que no aportarán valor real al equipo de trabajo y que normalmente son seleccionadas por tener presupuesto para invertir más no para mejorar las habilidades de las personas⁶⁸. Por lo tanto el plan de capacitación debe enfocarse

⁶⁵ BIEBERSTEIN, N. BOSE, S. FIAMMANTE, M. JONES, K. SHAH, R. Service-Oriented Architecture (SOA) Compass - Business Value, Planning, and Enterprise Roadmap. IBM Developer Works. 2006. p. 8.

⁶⁶ ALLEN, P. SOA Governance: Challenge or Opportunity? CBDI Journal. 2008. p. 13.

⁶⁷ CASTALDINI, F. SOA Governance and CentraSite Ensuring SOA success with effective, automated control throughout the lifecycle. Software AG. 2010. p. 9.

⁶⁸ BIEBERSTEIN, op. cit, p. 11.

en generar valor frente a los objetivos del Gobierno SOA teniendo en cuenta los siguientes puntos:^{69 70 71}

- ¿Qué habilidades es necesario mejorar con respecto a las características de cada rol?
- ¿Cuánto tiempo se dedicará mensualmente a las capacitaciones con respecto al plan de trabajo?
- ¿Qué capacitaciones cubren las habilidades que son necesarias mejorar?
- ¿Qué seguimiento se realizará a cada persona que ha recibido una o más capacitaciones después de que estas terminen?

4.1.8 Establecimiento de comités. La conformación de uno o más comités se soporta en las necesidades de la Arquitectura Orientada a Servicios y representan no solo la visión de SOA sino la visión de la organización⁷². La conformación de dichos grupos permite canalizar tanto los Principios SOA como las Políticas que estén encaminadas a alcanzar los objetivos que se han trazado⁷³.

El o los comités articulan y promueven la participación e integración de los diferentes equipos de trabajo que se encuentran desarrollando una Arquitectura Orientada a Servicios⁷⁴. Estas estructuras pueden ser virtualizadas mediante la

⁶⁹ Ibid., p. 8.

⁷⁰ ALEXANDER, Christopher. A Pattern Language. Oxford University Press. 2003. p. 1.

⁷¹ FARZIN, YASHAR. ROBERT, LAIRD. SOA and the Emergence of Business Technology: How Business Services are changing the IT Landscape. 2012. p. 12.

⁷² MARKS, E. Service-Oriented Architecture Governance for the Services Driven Enterprise. New Jersey: John Wiley & Sons Inc. 2008. p. 4.

⁷³ HEVNER, A.R. MARCH, S.T. PARK, J. RAM, S. Design science in information systems research. Management Information Systems Quarterly. Vol. 28. 2004. p. 6.

⁷⁴ SPENCER, L. RITCHIE, J. LEWIS, J. DILLON, L. Quality in Qualitative Evaluation: A framework for assessing research evidence. Government Chief Social Researcher's Office. 2003. p. 3.

asignación parcial o temporal de sus diferentes miembros, lo cual dependerá del tamaño de la organización y del alcance de los proyectos⁷⁵.

4.1.9 Establecimiento de Procesos. La selección y definición de los procesos necesarios para el establecimiento de un Gobierno SOA (Procesos de Gobierno SOA⁷⁶), debe ser realizada sin olvidar la estructura organizacional, debido a que cada uno de ellos deberá aportar y no entorpecer el flujo continuo de trabajo⁷⁷.

Existen varios métodos de trabajo que buscan implantar un Gobierno SOA, sin embargo deberá realizarse previamente una selección para determinar cuáles de ellos son aplicables y en qué momento⁷⁸. No es lógico pensar que la Organización se adapte al método, es más lógico seleccionar herramientas de una o varias fuentes con el fin de alcanzar el objetivo deseado, el cual se centra en el establecimiento de un Gobierno SOA, en pro de una Arquitectura Orientada a Servicios⁷⁹.

4.1.10 Diseño del ciclo de vida de los servicios. Los ciclos de vida definen las fases por las que pasa un servicio mientras esta activo en la SOA. Se pueden entender como “peajes” naturales donde se aplican políticas⁸⁰. Si se combinan los ciclos de vida con las políticas se genera un gobierno flexible y puntos de control⁸¹. Los arquitectos buscan automatizar el ciclo de vida SOA, para verificar

⁷⁵ WEILL, P. ROSS, J.W. IT Governance. Harvard Business Press. 2004. p. 11.

⁷⁶ DUPUIS, R. A Design Science Approach To Develop A New Comprehensive SOA Governance Framework. IEEE Press. 2001. p. 10.

⁷⁷ SESSIONS, R. CLARK, J. MANDER, K. A Framework For Evaluating SOA Governance Frameworks. ACM SIGSOFT Software Engineering Notes. Vol. 23. 2005. p. 2.

⁷⁸ JHONSON, J. An Integrated Model For SOA Governance. Journal of Systems and Software. Vol. 83. 2010. p. 8.

⁷⁹ HARMAN, M. ServiceOriented Architecture and Web Services. Department of Computer Science, King's College London, Tech. Rep. TR-09-03, 2009. p. 13.

⁸⁰ MOSCATO, P. Developing A Framework For Evaluating Service Oriented Architecture Governance SOAG. Caltech concurrent computation program. 2006. p. 7.

⁸¹ SOTELO, S. A Systematic Mapping Study On Non Functional Search Based Software Testing. Software Quality Journal. Vol. 19. 2011. p. 15.

constantemente resultados lo más exactos posibles y publicar resúmenes en el Registro/Repositorio⁸².

Los servicios no aparecen de forma instantánea y después existen para siempre. Como cualquier software, necesitan ser planificados, diseñados, implementados, desplegados, mantenidos y, finalmente, retirados del servicio. La aplicación del ciclo de vida puede ser pública y afectar muchas partes de una organización, pero un ciclo de vida del servicio puede tener un impacto más grande porque muchas aplicaciones dependen de un solo servicio⁸³.

El ciclo de vida de los servicios se hace más evidente cuando considera el uso de un registro. Es entonces cuando surge la necesidad de responder interrogantes como:^{84, 85, 86}

- ¿Cuándo debe agregarse un nuevo servicio al registro?
- ¿Todos los servicios en un registro se encuentran necesariamente disponibles y preparados para su uso?
- ¿Un servicio retirado debería eliminarse del registro?

Las dos partes, los consumidores y los proveedores, tienen que acordar cómo van a trabajar juntos. Gran parte de este conocimiento puede capturarse en un acuerdo de nivel de servicios (ANS), metas medibles que un proveedor de

⁸² HOHPE, G. Estrategia Dirigida Por Modelo Para El Gobierno SOA. Springer. 2007. p. 4.

⁸³ MUJTABA, S. MATTSSON, M. Structuring SOA Solutions From A Vision Enterprise Architecture. 12th International Conference on Evaluation and Assessment in Software Engineering. 2008. p. 3.

⁸⁴ GRIMÁN, A. JURISTO, N. Governance Maturity Tooling Vitality And Success Patterns. Empirical Software Engineering. Vol. 14. 2009. p. 8.

⁸⁵ FRED, D. User Acceptance of Computer Technology: A comparison of Two Theoretical Models. EEUU Vol 35. 2009. p. 14.

⁸⁶ AFSHAR, M. SOA Governance: Optimizes The Bussiness And Evolution Of Service Oriented Systems. Oracle. 2009. p. 10.

servicios acuerda alcanzar y con el que acuerda convivir un consumidor de los servicios. Este acuerdo es como un contrato entre las partes y puede, de hecho, ser un contrato legal. En el mejor de los casos, el ANS articula lo que el proveedor debe hacer y lo que espera el consumidor⁸⁷.

Una vez que el CEO del Gobierno aplica las políticas, puede utilizarse tecnología para gestionar esas políticas. La tecnología no define un ANS, pero puede utilizarse para exigir y medir el cumplimiento⁸⁸. Por ejemplo, la tecnología puede limitar qué consumidores pueden invocar un servicio y cuándo pueden hacerlo. Puede advertir a un consumidor que ese servicio ya no se utiliza. Puede medir la disponibilidad del servicio y el tiempo de respuesta.

4.1.11 Diseño del Portafolio/Repositorio de Servicios. El portafolio de servicios provee una fuente de información común a los diferentes actores en una Arquitectura Orientada a Servicios. Este concentra la información relacionada con las funcionalidades de cada servicio, dependencias, estados o ciclos de vida, consumidores, proveedores y posibles mediadores⁸⁹.

Por lo tanto, basado en un ámbito económico a nivel organizacional, el portafolio de servicios está vinculado con el conjunto de actividades o acciones que realiza la organización, para responder a las necesidades de sus consumidores, independiente de si estos son internos o externos⁹⁰.

El portafolio de servicios contiene una relación ordenada de servicios pertenecientes a una o varias áreas funcionales, que por su estructura precisan de

⁸⁷ HURTADO, J. Governance Life Cycle: Information and Software Technology. The IBM Systems Journal. 2011. p. 11.

⁸⁸ KEPNER, T. Introduction To Enterprise IT And SOA Governance: an introduction. Springer. Vol. 4. 2000. p. 4.

⁸⁹ ENDREI, Mark. ServiceOriented Architecture and Web Services. EEUU: IEEE Xplore. 2004. p. 2.

⁹⁰ JACUZZI, E. Presenting A Model For The Deployment Of Service Oriented Architecture Governance In Information And Communication Technology Department Of Isfahan Municipality. IEEE DEST. 2009. p. 7.

esa catalogación para facilitar su ubicación. En gran medida es comparable con un diccionario, donde por cada concepto (en este caso un servicio) existe una explicación (en este caso una funcionalidad).

La información almacenada no es de carácter técnico únicamente, lo cual es de suma importancia para los diferentes equipos funcionales de la organización, que buscan servicios ya implementados con el fin de reutilizar un activo existente y no volverlo a crear por segunda vez⁹¹.

La reutilización potencial de servicios y activos existentes se basa en un sólido *Portafolio de Servicios*, el cual debe almacenar toda la información necesaria para determinar, si la funcionalidad registrada previamente puede o no suplir una necesidad vigente, mejorando el tiempo de respuesta e incrementando el ROI de la organización⁹².

4.1.11.1 Trabajando con el Portafolio/Repositorio de Servicios. El repositorio es el cimiento del sistema de gobierno. Por el pasan todos los elementos usados durante el desarrollo y facilita el descubrimiento de los servicios. Se define como un catálogo maestro donde se encuentran todos los servicios y metadatos relacionados con el desarrollo SOA para catalogación, localización y consumo de servicios. Contiene los elementos más importantes de una SOA:^{93 94 95}

- Esquemas XML y procesos BPEL.
- Organizaciones que proporcionan servicios.

⁹¹ HOJAJI, F. AYATOLLAHZADEH SHIRAZI, M. R. SOA Governance IBM Approach, Proceedings 2010 2nd IEEE. International Conference on Information Management and Engineering. 2005. p. 9.

⁹² AFSHAR. 2009, op. cit, p. 11.

⁹³ IEEE. 2000, op. cit, p. 11.

⁹⁴ ERL. 2005, op. cit, p. 162.

⁹⁵ NIEMANN, op. cit, p. 14.

- Políticas de gobierno del comportamiento de las personas y sistemas del ciclo de vida.
- Aplicaciones que consumen servicios.
- Organizaciones que consumen servicios.
- Contratos entre proveedor/consumidor.
- Dependencias y relaciones entre los anteriores elementos.

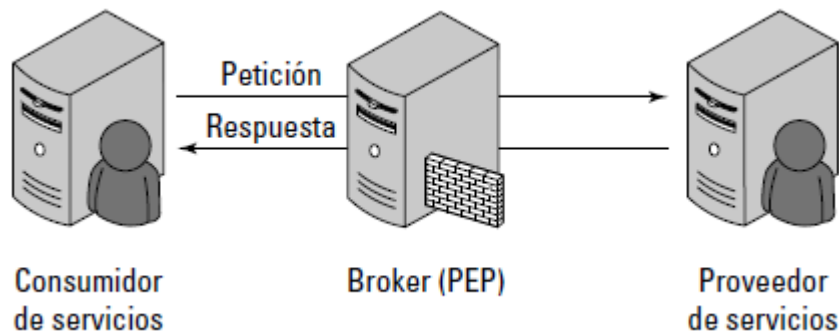
Si bien es cierto que el repositorio es el cimiento del sistema de gobierno, es importante tener en cuenta que este registro debe dar soporte para todos los perfiles involucrados y estar siempre accesible a cualquier tipo de consumidor⁹⁶. Debe ser compatible con entornos heterogéneos generando la posibilidad de funcionar en todas las plataformas tecnológicas. Por último el repositorio debe ser sensible a la personalización y a la capacidad de extensión facilitando la personalización de la información y adaptándola a las necesidades de la organización⁹⁷.

⁹⁶ ERL. 2007, op. cit, p. 343.

⁹⁷ MOORE, op. cit, p. 3.

4.1.11.2 Vinculación entre consumidores y servicios

Figura 6. Ejemplo de servicio virtual



Fuente: MATSUMURA Mike, BRULE Bjorn & SHAH Jingles⁹⁸.

En términos más prácticos, la dinámica para trabajar en una SOA se debería reflejar en los siguientes pasos:^{99, 100}

- El consumidor llega al registro/repositorio.
- El consumidor negocia un acuerdo de nivel de servicio (SLA) para garantizar seguridad por medio del acuerdo.
- El consumidor inicia un proceso para vincularse con un servicio a través de un punto de acceso.
- Se recibe una dirección de interfaz virtual que reside en un PEP.

Para finalizar el esquema de acceso a una SOA cabe mencionar que uno de los objetivos del desarrollo es realizar una fusión entre *gestión en tiempo de ejecución*

⁹⁸ MATSUMURA, op. cit, p. 47.

⁹⁹ ALLEN, P. The Influence Of SOA Governance Mechanisms On IT Flexibility And Service Reuse. CBDJ Journal. 2010. p. 9.

¹⁰⁰ ZACHMAN, op. cit, p. 6.

y el *repositorio* con el fin de garantizar el descubrimiento automático de servicios, la detección de dependencias entre servicios/consumidores de servicios, la publicación de información sobre rendimiento de servicios/gobierno en el repositorio y la detección de diferencias entre la información del repositorio y la ejecución¹⁰¹.

4.1.11.3 Registrando servicios en el Portafolio/Repositorio. ¿Cómo hacen los proveedores de servicios para tener disponibles sus servicios y que sean conocidos? ¿Cómo hacen los consumidores de servicios para ubicar los servicios que quieren invocar? Estas son responsabilidades del registro de servicio. Actúa como una lista de servicios disponibles y apunta a invocarlos.

El registro de servicio también ayuda a coordinar versiones de un servicio. Los consumidores y los proveedores pueden especificar qué versión necesitan o tienen y, entonces, el registro se asegura de solamente enumerar los proveedores de una versión deseada por el consumidor¹⁰². El registro puede administrar la compatibilidad de la versión, rastrear la compatibilidad entre versiones y enumerar los proveedores de una versión deseada por el consumidor o versiones compatibles¹⁰³. El registro también puede soportar estados de servicio, como pruebas, desaprobaciones, y hacer que solo estén disponibles los servicios con estos estados para los consumidores que los quieren.

Cuando un consumidor comienza a utilizar un servicio, se genera una dependencia a ese servicio. Mientras que cada consumidor sabe claramente de qué servicios depende, generalmente en una empresa, estas dependencias pueden ser difíciles de detectar y mucho menos de administrar¹⁰⁴. Un registro no

¹⁰¹ DAVIS, F. D. Towards A Service Portfolio Management Framework. Software AG. 7ma. Ed. Pearson Educación. S.A. 2010. p. 3.

¹⁰² MATSUMURA, op. cit, p. 70.

¹⁰³ DAVIS, op. cit, p. 10.

¹⁰⁴ JACUZZI, op. cit, p. 4.

solo puede generar una lista de servicio y proveedores, sino que también puede rastrear dependencias entre los consumidores y los servicios¹⁰⁵. Este rastreo puede ayudar a responder la pregunta ¿Quién utiliza el servicio? Entonces el conocimiento del registro de dependencias puede notificar a los consumidores sobre los cambios en los proveedores, por ejemplo, cuándo un servicio es desaprobado.

Un buen lugar para exigir el cumplimiento de las políticas de Gobierno es mediante una combinación de un bus de servicios empresariales (ESB) y un registro de servicios. Puede exponerse un servicio con el fin de que solo ciertos ESB puedan invocarlo¹⁰⁶. Entonces, la combinación ESB/registro puede controlar el acceso de los consumidores, supervisar y medir el uso, medir el cumplimiento del ANS y demás. De esta forma, los servicios se concentran en la proporción de funcionalidad comercial y el ESB/registro se centra en los aspectos del Gobierno.

4.2 PREGUNTA DE INVESTIGACIÓN 2

¿Cuáles son los criterios de selección más importantes para los arquitectos e ingenieros de software al momento de decidirse por una herramienta de gobierno que permita la gestión adecuada del portafolio de servicios?

4.2.1 Clasificación de las características de una herramienta adecuada para el Gobierno SOA. Existe la tendencia de dividir el Gobierno, y por lo tanto las herramientas utilizadas, en dos fases del ciclo de vida de una solución SOA, las herramientas en tiempo de diseño (DTG - Design Time Governance Tools) y las herramientas en tiempo de ejecución (RTG - Run Time Governance Tools)¹⁰⁷. Las DTG tienen por principal objetivo la gestión de los artefactos y el ciclo de vida de los servicios (por ejemplo registros y repositorios) mientras que las RTG son las

¹⁰⁵ HOJAJI, op. cit, p. 10.

¹⁰⁶ ENDREI, op. cit, p. 3.

¹⁰⁷ NIEMANN, op. cit, p. 12.

encargadas de garantizar la ejecución y el correcto funcionamiento de los servicios (conforme a las políticas y SLA definidos) así como el proporcionar métricas de comportamiento (utilizando cuadros de mando - BAM)¹⁰⁸.

Las características o funcionalidades que debe proporcionar una herramienta de Gobierno SOA se pueden clasificar en tres grandes grupos¹⁰⁹:

- Registro y Repositorio. el nivel mínimo de Gobierno SOA es poder ser capaz de conocer qué hay que gobernar y dónde se encuentran estos elementos. Por lo tanto, el registro y repositorio de todos los artefactos involucrados en un Gobierno SOA (no sólo de los servicios) debe ser la funcionalidad principal de la herramienta¹¹⁰.
- Gestión del Ciclo de Vida de los Servicios. El Gobierno SOA debe poder gestionar de forma completa el ciclo de vida de los servicios. Este ciclo de vida se puede describir como el conjunto de actividades relacionadas con las diferentes fases por las que pasa un servicio, desde su identificación hasta su implantación, sabiendo que este ciclo de vida se caracteriza por ser muy rápido y dinámico (a diferencia del ciclo de vida de las aplicaciones)¹¹¹.
- Administración de Políticas, Métricas, Monitorización y Configuración. El gobierno no es posible sin la definición y el seguimiento de un conjunto de políticas o reglas que definan y delimiten el comportamiento esperado. Así mismo el poder monitorizar lo que ocurre en nuestro ecosistema y configurar la herramienta según nuestras necesidades es un valor añadido a tener en cuenta¹¹².

¹⁰⁸ AYATOLLAHZADEH SHIRAZI, op. cit, p. 5.

¹⁰⁹ JHONSON, op. cit, p. 2.

¹¹⁰ HARMAN, op. cit, p. 10.

¹¹¹ HURTADO, op. cit, p. 7.

¹¹² SOTELO, op. cit, p. 1.

4.2.2 Características deseables en una herramienta adecuada para el Gobierno SOA. Basándonos en la clasificación anterior, algunas características o funcionalidades que debería poder realizar una herramienta de Gobierno SOA adecuada son las siguientes:

- **Registro y Repositorio^{113, 114, 115}**
 - Implementar un registro de artefactos.
 - Poder acceder al registro de servicios de forma programática, es decir, disponer de un API para el acceso.
 - Implementar mecanismos de dependencia de unos servicios con otros.
 - Permitir añadir información adicional (metadatos) a los servicios registrados.
 - Proporcionar mecanismos para estructurar y organizar el conjunto de servicios.
 - Proporcionar mecanismos ágiles de búsqueda dentro del repositorio.
- **Gestión del Ciclo de Vida de los Servicios^{116, 117}**
 - Gestionar el ciclo básico de servicios (análisis/diseño, desarrollo, pruebas y explotación).

¹¹³ MOSCATO. 2006, op. cit, p. 14.

¹¹⁴ GRIMÁN, op. cit, p. 11.

¹¹⁵ HARMAN, op. cit, p. 8.

¹¹⁶ FRED, op. cit, p. 6.

¹¹⁷ ENDREI, op. cit, p. 2.

- Administrar distintas versiones de los servicios.
- Integrar un workflow que permita la promoción de los servicios, es decir, que permita la aprobación por parte de un arquitecto SOA del paso de un servicio de diseño a desarrollo, de desarrollo a pruebas, etc.
- Integrar herramientas para la ejecución de pruebas de servicios.
- **Administración de Políticas, Métricas, Monitorización y Configuración¹¹⁸**
 - Implementación de diversas políticas estándares relacionadas con servicios web (por ejemplo el cumplimiento del Basic Profile 1.1).
 - Posibilidad de añadir o configurar nuevas políticas.
 - Proporcionar indicadores o métricas para monitorizar los servicios en ejecución (logs, auditoría, tiempos de respuesta, etc.).
 - Proporcionar una interfaz gráfica que muestre dichas métricas de forma amigable.
 - Notificar la ocurrencia de distintos eventos mediante el envío de mensajes (email, RSS, SMS, etc.).

Así mismo, desde el punto de vista comercial, se deberían tener en cuentas otros criterios:^{119 120 121}

¹¹⁸ GRIMÁN, op. cit, p. 7.

¹¹⁹ ERL. 2007, op. cit, p. 347.

¹²⁰ MOORE, op. cit, p. 10.

- Disponibilidad de Edición Community/Enterprise.
- Tipo de Licencia.
- Coste de Edición Community/Enterprise.
- Calidad de la Documentación.
- Soporte Multi-Idioma.
- Disponibilidad del Código Fuente.
- Comunidad activa.
- Coste del soporte.
- Disponibilidad de empresas que den el soporte de la herramienta.

4.3 AMENAZAS A LA VALIDEZ

Los resultados de un estudio de mapeo pueden ser afectados por los investigadores que realizan la investigación, por las bibliotecas digitales seleccionadas, por los términos de búsqueda y por la ventana de tiempo seleccionada¹²². Naturalmente, esto plantea una serie de amenazas a la validez de los resultados que son analizadas en la presente sección.

¹²¹ ALLEN. 2010, op. cit, p. 13.

¹²² ELBERZHAGER, op. cit, p. 9.

4.3.1 Validez de conclusión. La validez de conclusión se refiere a la relación entre el tratamiento y el resultado¹²³. Una amenaza a la validez de conclusión se puede dar en la extracción de datos. Para disminuir esta amenaza se desarrolló un formulario de extracción de datos que asegura que los datos relevantes extraídos sean consistentes.

4.3.2 Validez del constructo. La validez del constructo asegura que la construcción del estudio esté relacionada con el problema de investigación y que las fuentes seleccionadas sean relevantes¹²⁴. Una posible amenaza a la validez del constructo puede ser el sesgo en el proceso de selección de estudios primarios. Esta amenaza se disminuye con la construcción de un protocolo de investigación. Con el objetivo de obtener la máxima cantidad de estudios primarios relevantes, la cadena de búsqueda se construyó en forma sistemática. Los términos del tema de interés en primera instancia se obtuvieron de algunas sugerencias de términos de revisiones bibliográficas anteriores y se fueron ajustando durante las búsquedas preliminares realizadas.

4.3.3 Validez interna. La validez interna se refiere a la conexión entre la observación y las estrategias propuestas¹²⁵. Esto asegura que las conclusiones extraídas sean verdaderas. Esta amenaza se disminuye con un proceso definido de selección de artículos. La inclusión y exclusión de artículos se realizó en varias sesiones con el criterio de algunos arquitectos de software con experiencia en SOA, lo que disminuye la amenaza de la validez interna.

4.3.4 Validez externa. La validez externa se refiere a la posibilidad de generalización de los resultados¹²⁶. En la presente revisión se consideran solo estudios a partir del año 2006, lo que puede ser una amenaza para la

¹²³ WOHLIN. 2006, op. cit, p. 3.

¹²⁴ Ibid., p. 4.

¹²⁵ Ibid., p. 9.

¹²⁶ Ibid., p. 7.

generalización de los resultados. Para disminuir esta amenaza, los resultados se han comparado y complementado con los de otras revisiones similares. Por otra parte, dado el procedimiento sistemático seguido durante el estudio de mapeo, los resultados y las conclusiones generales derivadas de ellos tienen validez dentro del dominio de investigación abordado, permitiendo que los resultados del estudio puedan ser usados como un punto de partida para otras líneas de investigación en SOA.

4.4 DISCUSIÓN

Después de revisar los dos artículos primarios que guardaban más relación de similitud con esta investigación *“The Influence Of SOA Governance Mechanisms On IT Flexibility And Service Reuse”* y *“Gobierno SOA: Elemento Clave en la Integración de Negocio y Tecnología”*, se determinó que la principal contribución del presente estudio es la definición de las estrategias más adecuadas para la conformación de un Gobierno SOA y su portafolio de servicios, ya que si bien en dichos estudios se mencionan buenas prácticas para el gobierno y algunas recomendaciones, no se evidencian esquemas de clasificación ni de validez hacia el uso de esas prácticas.

Allen concluye que el Gobierno SOA resuelve el problema de la integración transversal de componentes atacándolo de raíz, ya que reemplaza el acoplamiento fuerte de las aplicaciones por un conjunto de servicios independientes vinculados a un portafolio de servicios¹²⁷. Nuestro estudio logra reafirmar dicha conclusión ya que identificamos que SOA busca encapsular la implementación interna de cada servicio, lo cual facilita enormemente su reutilización. Si se le agrega a esto la facilidad de identificar procesos complejos a partir de los servicios individuales, se puede entender por qué esta arquitectura puede ofrecer la flexibilidad requerida.

¹²⁷ ALLEN. 2010, op. cit, p. 15.

Bastida dentro de sus recomendaciones propone que el primer paso de un Gobierno SOA debería ser la integración de los procesos del negocio para encontrar todas las características del mismo (flujo, información requerida en cada paso, aplicaciones participantes, etcétera) ¹²⁸. Sin embargo, nuestro estudio complementa dicha propuesta con la importancia de la identificación de servicios en las aplicaciones existentes. Esto puede ser relativamente sencillo o bastante complejo, dependiendo del diseño de dichas aplicaciones. Es por esta razón que la gestión adecuada del portafolio de servicios garantiza la identificación de los servicios de una organización, haciéndolos visibles, manteniéndolos y publicándolos para ser consumidos por otras aplicaciones o clientes.

Este estudio adicionalmente proporciona a los profesionales una visión general de los problemas existentes y de los retos que sugiere solucionar cada uno de ellos a través de las estrategias definidas, entendiendo en qué contexto puede ser más eficiente aplicarlas. Para los investigadores, proporciona una visión general de los distintos enfoques para asumir adecuadamente el Gobierno SOA con el fin de identificar oportunidades en la investigación y poder establecer futuras líneas de investigación en SOA.

Finalmente otro aporte que realiza este estudio a la comunidad de desarrollo es la disponibilidad de acceder a estas estrategias de manera libre, lo que permite a cualquier arquitecto de software en proceso de adoptar una SOA lograr su objetivo sin la necesidad de tener que incurrir en gastos de costosas licencias de las empresas líderes en la industria SOA. Sabemos que dichas compañías tienen años de experiencia en SOA y seguramente sus estrategias son mucho más consistentes y efectivas, pero consideramos que la comunidad libre de desarrollo debe tener un punto de partida para darle a sus implementaciones la dirección adecuada.

¹²⁸ BASTIDA, op. cit, p. 14.

5. CONCLUSIONES Y TRABAJO FUTURO

5.1 CONCLUSIONES

El estudio permitió brindar al arquitecto o ingeniero de software una serie de estrategias para la conformación de un Gobierno y su portafolio de servicios en una implementación SOA, ofreciendo así una solución de calidad ante la presencia de algunos de los problemas más comunes que se presentan en la conformación de un Gobierno SOA y su portafolio de servicios.

La industria y los autores académicos tomados como referencia en este estudio están de acuerdo en afirmar que las compañías fallarán en la implantación de una SOA si no implementan de forma efectiva y sólida un Gobierno SOA. La complejidad en la implantación de un proyecto SOA ya no radica en el modelo tecnológico, donde actualmente ya se han desarrollado varios estándares y soluciones, sino en facilitar a los departamentos de IT la gestión integral del ciclo de vida de los servicios y la implantación de las políticas establecidas.

Se hace por lo tanto indispensable, además de un correcto enfoque metodológico, el poder contar con herramientas adecuadas para la implantación de los objetivos que establece un Gobierno SOA. No es posible disponer de una única herramienta que proporcione todas las características necesarias para el Gobierno SOA.

Mediante la evaluación que se ha realizado en el caso de estudio de las herramientas seleccionadas se puede concluir que ninguna herramienta cumple totalmente con la funcionalidad que se espera de una herramienta de Gobierno, aunque posean algunas ventajas frente a las otras, por lo que se hace necesario optar por adquirir varias herramientas que satisfagan la mayoría de las necesidades de una organización.

5.2 TRABAJO FUTURO

Se propone realizar una ampliación de las estrategias para la conformación de un Gobierno SOA y su portafolio de servicios con el fin de tener un catálogo de problemas específicos y estrategias de solución mucho más completo que se adapte mejor a un mayor número de situaciones comunes en las organizaciones.

Se propone también realizar un esquema de clasificación de las estrategias brindadas como más eficaces o eficientes para asumir los retos que surgen en la construcción de un Gobierno SOA. Dicha clasificación se podría basar en el método de implementación de las estrategias obtenidas y en la identificación del tipo de software con el que se está llevando a cabo.

También se recomienda la inclusión de ejemplos de la aplicación de las estrategias para la conformación de Gobierno SOA, con el fin que los ingenieros y arquitectos tengan una base en donde validen la forma correcta su implementación.

Se recomienda desarrollar un software (o un plugin de la herramienta de gobierno seleccionada) que realice la aplicación de las estrategias adecuadas para la construcción, mantenimiento y evolución del portafolio de servicios de forma automática de acuerdo a los requerimientos del proyecto, generando probabilidades de éxito para seleccionar la opción más apropiada a la situación en específico. Esto reduciría el tiempo de desarrollo y minimizaría los errores en la implementación de las mismas.

BIBLIOGRAFÍA

ABRAN, A. MOORE, J. W. Guide to the software engineering body of knowledge (swebok). IEEE Press. 2004.

AFSHAR, M. SOA Governance: Framework and Best Practices. Oracle. 2007.

AFSHAR, M. SOA Governance: Optimizes The Bussiness And Evolution Of Service Oriented Systems. Oracle. 2009.

ALEXANDER, Crhistopher. A Pattern Language. Oxford University Press. 2003.

ALLEN, P. SOA Governance: Challenge or Opportunity? CBDI Journal. 2008.

ALLEN, P. The Influence Of SOA Governance Mechanisms On IT Flexibility And Service Reuse. CBDI Journal. 2010.

AMBERPOINT. Actional Solutions for SOA Governance [Online]. 2015. Disponible en: <<http://www.actional.com/solutions/soa-governance/>>

ANDREU, Rafael. RICART, Joan. VALOR, Josep. Estrategia y Sistemas de Información. Mc Graw Hill. 1996.

AYATOLLAHZADEH SHIRAZI, M. R. Developing a More Comprehensive and Expressive SOA Governance Framework. China: 2nd IEEE International Conference on Information Management and Engineering. 2011.

BALUDA, M. BRAIONE, P. DENARO, G. PEZZÈ, M. Enhancing structural software coverage by incrementally computing branch executability. Software Quality Journal. Vol. 19. 2011.

BARRAZA, Fernando. Modelado y Diseño de Arquitectura de Software [Online]. 2012. Disponible en: <http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:s2_conceptosdemodelado.pdf>

BASTIDA, Leire. Gobierno SOA: Elemento Clave en la Integración de Negocio y Tecnología. Sevilla: JSWEBOCT08. 2008.

BIEBERSTEIN, N. BOSE, S. FIAMMANTE, M. JONES, K. SHAH, R. Service-Oriented Architecture (SOA) Compass - Business Value, Planning, and Enterprise Roadmap. IBM Developer Works. 2006.

BROWN, W. SOA Governance - IBM's Approach. IBM Express. 2006.

CASTALDINI, F. SOA Governance and CentraSite Ensuring SOA success with effective, automated control throughout the lifecycle. Software AG. 2010.

DAVIS, F. D. Towards A Service Portfolio Management Framework. Software AG. 7ma. Ed. Pearson Educación. S.A. 2010.

DEEPAL, Jayasinghe. Open Source Competition - Mule Galaxy vs WSO2 Registry. WSO2 Portal. 2008.

DIESTE, O. GRIMÁN, A. JURISTO, N. Developing search strategies for detecting relevant experiments. Empirical Software Engineering. Vol. 14. 2009.

DUPUIS, R. A Design Science Approach To Develop A New Comprehensive SOA Governance Framework. IEEE Press. 2001.

DURVASULA, Surekha. GUTTMANN, Martin. KUMAR, Ashok. LAMB, Jeffery. MITCHELL, Tom. ORAL, Burc. PAI, Yogish. SEDLACK, Tom. SHARMA, Harsh. SUNDARESAN, Sankar Ram. SOA Practitioners Guide. Part 3. Introduction to Services Lifecycle [Online]. 2006. Disponible en: <<http://www.soablueprint.com/whitepapers/SOAPGPart3.pdf>>

ELBERZHAGER, F. MÜNCH, J. NHA, V. T. N. A systematic mapping study on the combination of static and dynamic quality assurance techniques. Information and Software Technology. 2011.

ENDREI, Mark. ServiceOriented Architecture and Web Services. EEUU: IEEE Xplore. 2004.

ERL, T. Service Oriented Architecture - Concepts, Technology, and Design. Boston: Prentice Hall Professional Technical Reference. 2005.

ERL, T. SOA Governance Thomas Erl. Boston: Prentice Hall Professional Technical Reference. 2007.

FARZIN, YASHAR. ROBERT, LAIRD. SOA and the Emergence of Business Technology: How Business Services are changing the IT Landscape. 2012.

FRED, D. User Acceptance of Computer Technology: A comparison of Two Theoretical Models. EEUU. Vol. 35. 2009.

GRIMÁN, A. JURISTO, N. Governance Maturity Tooling Vitality And Success Patterns. Empirical Software Engineering. Vol. 14. 2009.

HARMAN, M. ServiceOriented Architecture and Web Services. Department of Computer Science, King's College London, Tech. Rep. TR-09-03, 2009.

HEVNER, A.R. MARCH, S.T. PARK, J. RAM, S. Design science in information systems research. Management Information Systems Quarterly. Vol. 28. 2004.

HOPPE, G. Estrategia Dirigida Por Modelo Para El Gobierno SOA. Springer. 2007.

HOJAJI, F. AYATOLLAHZADEH SHIRAZI, M. R. SOA Governance IBM Approach, Proceedings 2010 2nd IEEE. International Conference on Information Management and Engineering. 2005.

HURTADO, J. Governance Life Cycle: Information and Software Technology. The IBM Systems Journal. 2011.

IEEE. Recommended Practice for Architectural Description for Software-Intensive Systems, IEEE Standard 1471-2000, 2000.

IEEE. Advancing Technology for Humanity [Online]. 2015. Disponible en: <<https://www.ieee.org/about/index.html>>

JACUZZI, E. Presenting A Model For The Deployment Of Service Oriented Architecture Governance In Information And Communication Technology Department Of Isfahan Municipality. IEEE DEST. 2009.

JHONSON, J. An Integrated Model For SOA Governance. Journal of Systems and Software. Vol. 83. 2010.

KEPNER, T. Introduction To Enterprise IT And SOA Governance: an introduction. Springer. Vol. 4. 2000.

KERRIE, Holley. ALI, Arsanjani. 100 SOA Questions Asked and Answered. Boston Massachusetts: Prentice Hall. 2011.

LAKHOTIA, K. MCMINN, P. HARMAN, M. An empirical investigation into branch coverage for C programs using CUTE and AUSTIN. Journal of Systems and Software. Vol. 83. 2010.

LEWIS, Grace. SMITH, A. Dennis, B. KONTOGIANNIS, Kostas. A Research Agenda for Service-Oriented Architecture (SOA): Maintenance and Evolution of Service-Oriented Systems. Software Engineering Institute. 2010.

MAHMOUD, Qusay. Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI) [Online]. Oracle, 2005. Disponible en: <<http://www.oracle.com/technetwork/articles/javase/soa-142870.html>>

MANSOURI, S. A. ZHANG, Y. Search based software engineering: A comprehensive analysis and review of trends techniques and applications. Department of Computer Science, King's College London, Tech. 2009.

MARKS, E. Service-Oriented Architecture Governance for the Services Driven Enterprise. New Jersey: John Wiley & Sons Inc. 2008.

MATSUMURA Mike, BRULE Bjorn & SHAH Jingles. Adopción de SOA para dummies. Indiana: Wiley Publishing Inc., 2009.

MICROSOFT CORPORATION. Application Architecture Guide [Online]. 2009. Disponible en: <<https://msdn.microsoft.com/en-us/library/ff650706.aspx>>

MICROSOFT CORPORATION. Software Architecture and Design [Online]. 2012. Disponible en: <msdn.microsoft.com/en-us/library/ee658093.aspx>

MOSCATO, P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program. Vol. 826. 2009.

MOSCATO, P. Developing A Framework For Evaluating Service Oriented Architecture Governance SOAG. Caltech concurrent computation program. 2006.

MOORE, G. TEGAN, W. Systematic Mapping Studies In Software Engineering. Boston: IEEE Xplore. 2005.

MUJTABA, S. MATTSSON, M. Structuring SOA Solutions From A Vision Enterprise Architecture. 12th International Conference on Evaluation and Assessment in Software Engineering. 2008.

MULE GALAXY. Mule Galaxy: Mule Source's Open Source SOA [Online]. 2014. Disponible en: <www.mulesource.org/display/GALAXY/Home>

NIEMANN, M. CHRISTIAN, J. NICOLAS, R. RALF, S. Challenges of Governance Approaches for Service-Oriented Architectures. IEEE. 2009.

PETERSEN, K. FELDT, R. Systematic Mapping Studies in Software engineering. Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. 2008.

RESEARCH IBM. Nanotechnology & Nanoscience [Online]. 2013. Disponible en: <<http://www.research.ibm.com/index.shtml>>

SADTLER, Carla. HUBER, Philipp. YI, SangMin. Enabling SOA Using WebSphere Messaging. IBM Redbooks. 2006.

SESSIONS, R. CLARK, J. MANDER, K. A Framework For Evaluating SOA Governance Frameworks. ACM SIGSOFT Software Engineering Notes. Vol. 23. 2005.

SPENCER, L. RITCHIE, J. LEWIS, J. DILLON, L. Quality in Qualitative Evaluation: A framework for assessing research evidence. Government Chief Social Researcher's Office. 2003.

SOA BLUEPRINT. About US, Strategic Use of IT [Online]. Disponible en: <http://www.soablueprint.com/about_us>

SOTELO, S. A Systematic Mapping Study On Non Functional Search Based Software Testing. Software Quality Journal. Vol. 19. 2011.

TRACEY, N. CLARK, J. MANDER, K. Automated program flaw finding using simulated annealing. ACM SIGSOFT Software Engineering Notes. Vol. 23. 2008.

UNIVERSIDAD DE SEVILLA. Introducción a la arquitectura del Software [Online]. 2013. Disponible en: <www.lsi.us.es/docencia/get.php?id=6496>

WEILL, P. ROSS, J.W. IT Governance. Harvard Business Press. 2004.

WIKIPEDIA. Oracle Corp [Online]. 2015. Disponible en: <https://es.wikipedia.org/wiki/Oracle_Corporation>

WOHLIN, C. Experimentation in software engineering: an introduction. Springer. Vol. 6. 2000.

WOHLIN, C. Experimentation in software engineering: Service-Oriented Architecture. Springer. Vol. 1. 2006.

ZACHMAN, John. A framework for Information Systems Architecture. The IBM Systems Journal vol. 26. 1987.

ANEXOS

ANEXO A. IMPLEMENTACIÓN DE LAS ESTRATEGIAS EN EL CASO DE ESTUDIO

La industria es consciente de la importancia que tiene el establecer un buen Gobierno SOA para llevar a cabo con éxito una estrategia de implantación de proyectos con un enfoque orientado a servicios. Para conseguirlo, una pieza fundamental es poder integrar el ciclo de vida de los servicios en la metodología de desarrollo y despliegue de aplicaciones. El no haber integrado de forma efectiva el Gobierno SOA dentro del ciclo de vida del software puede ser una de las causas de una mala adopción de SOA.

Así como las herramientas de gestión de la configuración (CVS, SVN, etc.) han facilitado la gestión integral de los componentes de desarrollo (clases, librerías, componentes e incluso aplicaciones) se necesitan nuevas herramientas para gestionar el Gobierno SOA, sus políticas, métricas, roles y los servicios.

En esta etapa del proyecto de investigación se realizó la implementación de las estrategias para la adecuada gestión del portafolio de servicios en una reciente conformación de Gobierno en un caso de estudio particular de una empresa desarrolladora de software Colombiana ubicada en la ciudad de Bogotá donde el estudiante autor de este trabajo inicio su práctica empresarial en el área de calidad y al día de hoy es programador Junior.

Contexto: La empresa objetivo del caso de estudio hace unos años realizo un intento por adoptar una arquitectura orientada a servicios sobre su arquitectura IT tradicional dado que empezaron a presenciar múltiples problemas comunes de este tipo de empresas, algunos de ellos enunciados anteriormente en el desarrollo

de este documento. La iniciativa empezó muy bien y obtuvo buenos resultados al principio, pero después de indagar un poco sobre el suceso se pudo evidenciar un punto de quiebre donde la empresa dejó de creer en el paradigma y regreso a sus antiguos procesos “problemáticos”.

Ese punto estaba asociado a un problema con el Gobierno de la SOA, específicamente con el manejo adecuado del Portafolio de servicios, ya que de un momento a otro se presentaron varios problemas que impedían que la SOA evolucionara de manera adecuada asociados a la gestión del portafolio.

Con el fin de estudiar la influencia de las estrategias generadas en la metodología de investigación en una situación real, la empresa desarrolladora de software ha decidido ofrecerse como caso de estudio para que el equipo de arquitectos de software evalúen herramientas Open Source para implantar un Gobierno SOA de acuerdo con las estrategias propuestas en esta investigación, una de la compañía Mule, conocida como Galaxy, Membrane de la compañía predic8 y la tercera Governance Registry, de la compañía WSO2.

ANEXO B. EVALUACIÓN DE PLATAFORMAS OPEN SOURCE PARA EL GOBIERNO SOA

Hasta la aparición de Galaxy, las herramientas de Gobierno SOA Open Source se habían centrado en proporcionar un repositorio de servicios. Galaxy pretendía ir más allá, proporcionando funcionalidades adicionales para la gestión del Gobierno SOA.

En Julio de 2009, la compañía WSO2 ha contraatacado liberando la versión 3.0 de WSO2 Registry, a la cual se le ha cambiado el nombre a WSO2 Governance Registry (G-Reg) ya que añade nuevas funcionalidades relacionadas con el ciclo de vida de los servicios, la administración de políticas y la monitorización.

La evaluación realizada ha consistido en la descarga e instalación de las versiones seleccionadas y la realización de algunas pruebas que permitieran evaluar las funcionalidades identificadas.

ANEXO C. MULE – GALAXY

Registro y Repositorio. A la hora de implementar un registro de artefactos Galaxy no sólo permite el registro de servicios, sino de los siguientes artefactos:

- Documentos WSDL (permite el registro de servicios web).
- Políticas de servicios web.
- Archivos JAR.
- Configuraciones de Mule (el ESB de la misma compañía).
- Configuraciones de Spring.
- Esquemas XML (XSD).
- Hojas de estilo XSLT.

También permite registrar otros tipos de artefactos aunque no tenga soporte directo para los mismos. También facilita la extensión del sistema para añadir soporte específico a nuevas tipologías de artefactos. Un ejemplo de esta nueva tipología de artefactos podría ser la documentación asociada a un servicio web cuyo soporte incluido únicamente en la edición Enterprise.

- **Acceso programático al registro.** Galaxy permite acceder de forma programática a la herramienta de tres formas, mediante el uso de scripts Groovy, mediante el HTTP AtomPub API y mediante el API Java de Galaxy. Los scripts Groovy deben ser ejecutados desde la consola de administración de la herramienta. El API HTTP AtomPub está basado en un servicio RESTful que expone Galaxy para la interacción con clientes de la herramienta. El API de Galaxy da acceso a toda la herramienta. Está ideado para la extensión de la misma y para su integración con otras herramientas. A diferencia del acceso AtomPub, el API de Galaxy obliga a utilizar sus librerías (ficheros jar) por lo que su uso está reservado a aplicaciones Java. Una ventaja es que al ser Galaxy una aplicación Spring resulta sencillo alojar Galaxy o usar sus capacidades dentro de otra aplicación. En la siguiente captura se muestra cómo se accedería al Galaxy desde una aplicación tipo Spring.

Figura 7. Accediendo al Galaxy desde una aplicación tipo Spring

```
String configURL = "http://admin.admin@localhost:8002/api/registry?q=select artifact where spring bean = 'TestObject'";  
ApplicationContext context = new GalaxyApplicationContext(new URL(configURL));  
.... = context.getBean("bean");
```

Fuente: Autor.

- **Añadir información adicional (metadatos) a los servicios registrados.** Galaxy permite añadir meta información a los artefactos registrados. Esta meta información organizada en forma de propiedades, puede ser personalizada modificando las incluidas en la distribución original así como añadiendo nuevas propiedades para adaptar el sistema a las necesidades de la organización. Esta adaptación permite que un artefacto disponga de metadatos globales, con un único valor en un momento determinado, y metadatos versionados, con un valor distinto por cada versión de un artefacto.

- **Dependencia entre servicios.** Galaxy detecta y maneja la dependencia directa entre artefactos. En el caso de documentos WSDL, localiza las etiquetas import y crea una dependencia entre el artefacto del documento WSDL y los elementos referenciados en dichas etiquetas. Estos elementos serán habitualmente esquemas XSD. Es posible también añadir una dependencia de forma manual, las cuales, tienden a cubrir la relación funcional entre artefactos, esta relación no la puede detectar ni tratar directamente la herramienta, mientras que la dependencia generada por Galaxy cubre la relación técnica entre artefactos.
- **Organizar el conjunto de servicios.** Galaxy maneja el concepto de espacio de trabajo (workspace). Dentro de cada workspace, los artefactos se organizan por su tipología. La información mostrada por cada artefacto depende del tipo del mismo existiendo una propiedad que está presente en todos ellos, la versión. Es posible añadir vistas personalizadas que pueden incluir diversos criterios de selección que pueden ser especificados utilizando un asistente o bien introducirlos directamente a través del Galaxy Query Language (GQL). El GQL es un lenguaje similar a SQL como se aprecia en la sentencia de la siguiente captura.

Figura 8. Galaxy Query Language (GQL)

```
select artifact where wsdl.service = 'CustomerService';
```

Fuente: Autor.

- **Búsquedas en el repositorio.** Haciendo uso de las vistas, Galaxy permite localizar los artefactos con relativa facilidad. Además es posible añadir y gestionar índices para facilitar la búsqueda de servicios. Sin embargo, esta funcionalidad está disponible tan sólo en la edición Enterprise de la herramienta.

Gestión del Ciclo de Vida de los Servicios

- **Gestionar el ciclo básico de servicios.** Galaxy soporta el ciclo de vida básico de un servicio, el cual incluye las fases habituales del desarrollo de un servicio. Las definidas por defecto son: *Created, Developed, Tested, Production, Retired*. Tanto las fases como su secuencia son modificables de forma que podemos adaptar y personalizar este *workflow* a nuestras necesidades.

- **Integrar un *workflow* que permita la promoción de los servicios.** Galaxy integra un workflow (que puede ser modificado) que permite la promoción de los servicios, esto quiere decir que permite la aprobación por parte de un arquitecto SOA del paso de un servicio de diseño a desarrollo, de desarrollo a pruebas, etc. Sin embargo, a pesar de poder añadir nuevos roles a usuarios e incluso utilizar LDAP para la autenticación, no es posible indicar a la herramienta que algunas transiciones requieren de un perfil de usuario concreto. En este sentido Galaxy resulta incompleto.

- **Administrar versiones de los servicios.** Galaxy gestiona adecuadamente el versionado de los artefactos. Crear una nueva versión resulta sencillo. Es posible marcar la versión anterior como deshabilitada. En el caso de documentos WSDL es posible especificar una política de compatibilidad con versiones anteriores, de forma que la herramienta nos avisa si una nueva versión del documento resultara incompatible con la versión anterior.

Como ejemplo de la validación mencionada, en la siguiente captura se muestra el mensaje de error de Galaxy cuando intentamos registrar una nueva versión de un servicio sin una de las operaciones de la versión anterior.

Figura 9. Mensaje de error de Galaxy



Fuente: Autor.

- **Ejecución de pruebas de servicios.** Galaxy no tiene soporte para la ejecución de pruebas de servicios. A pesar de esta limitación, es posible añadir trabajos planificados para la ejecución desde la herramienta. Con este mecanismo, e implementando clases *ad-hoc*, se podría automatizar la ejecución de pruebas de servicios. También es muy sencillo integrar Galaxy con Mule ESB con lo que podríamos tener servicios de pruebas de servicios dentro del ESB e invocarlos de forma planificada desde Galaxy.

Políticas, Métricas, Monitorización y Configuración

- **Implementación de diversas políticas estándares.** Galaxy incluye algunas políticas en su distribución. Una de ellas es la que obliga, si se activa dicha política, a que los documentos WSDL cumplan la especificación WS-I Basic Profile 1.1. Otra política asociada a los documentos WSDL es la relativa a la compatibilidad entre versiones de este tipo de artefactos. También permite establecer qué políticas entran en acción según la fase del ciclo de vida de los servicios.
- **Adición y configuración de nuevas políticas.** Galaxy permite añadir nuevas políticas de forma programática (no desde la consola de administración), proporcionando un API para el desarrollo de políticas personalizadas.

- **Métricas para monitorizar los servicios en ejecución.** Galaxy conoce el estado de los servicios tan sólo como consecuencia de la modificación de la fase en la que se encuentran y no como resultado de ninguna prueba de ejecución o similar. Es el único indicador que se muestra en la herramienta. No obstante, Galaxy contiene un API relativo a eventos que permite registrar una traza de la actividad de la herramienta. Este API facilita el registro de *listeners* lo que nos permite reaccionar ante cualquier evento generado en el motor de la herramienta. Los *listeners* pueden registrarse indicando su forma de ejecución, es decir, síncrona y asíncrona.
- **Interfaz gráfica amigable para las métricas de monitorización.** Galaxy no proporciona ninguna interfaz gráfica específica para métricas ya que no las contempla como tal. La compañía tenía pensado incluir en su *roadmap* el lanzamiento de Mule Saturn, una herramienta BAM para la monitorización de su infraestructura SOA (ESB y Galaxy) aunque dicha herramienta aún no ha sido puesta a disposición al público.
- **Notificar la ocurrencia de distintos eventos.** Galaxy permite la sindicación (RSS) relativa a tres niveles: *workspace*, artefacto y comentarios. La sindicación de cada *workspace* notifica el alta o baja de artefactos. La sindicación de cada artefacto notifica cualquier modificación sobre el mismo, incluido su versionado. La sindicación de comentarios es global, cualquier comentario añadido a cualquier artefacto de cualquier *workspace* provocará una nueva entrada RSS. Aunque Galaxy registra todos los cambios realizados sobre un artefacto, el RSS tan sólo muestra el versionado y el momento en el que se ejecuta una modificación sobre el mismo.

ANEXO D. WSO2 - G-REG

Registro y Repositorio. G-Reg permite registrar y acceder de una forma cómoda a los servicios, Políticas, WSDL y Esquemas XSD. El registro de los servicios se realiza bien a través de un formulario o de una forma más cómoda importando el WSDL asociado. Así mismo permite el registro de otros tipos de recursos (texto plano, URLs, links simbólicos) o incluso recursos registrados en otros repositorios. La herramienta posee un método para definir tipos de recursos de forma que se puedan clasificar y agilizar la salida de los artefactos.

- **Acceso programático al registro.** G-Reg dispone de un API Java propio para el acceso a todos los elementos del repositorio. A su vez, este API se encuentra expuesto como servicio web. También proporciona distintos mecanismos para extender la funcionalidad de la herramienta como son handlers (permite ejecutar lógica extra asociada a la ocurrencia de eventos), filters (asocia los tipos de recursos a los handlers) y aspects (lógica extra a invocar por el usuario).
- **Añadir información adicional (metadatos) a los servicios registrados.** G-Reg al igual que Galaxy también permite añadir meta información a todos los artefactos registrados en forma de descripción textual, comentarios e incluso propiedades a medida.
- **Dependencia entre servicios.** G-Reg al registrar un documento WSDL o XSD, importa todos los documentos relacionados generando las dependencias entre ellos. Así mismo, también permite añadir dependencias entre otros artefactos de forma manual e incluso asociaciones, donde la dependencia no está ligada a un recurso sino a un tipo.

- **Organizar el conjunto de servicios.** G-Reg utiliza el concepto de colección como una agrupación de recursos. Además proporciona mecanismos para poder clasificar los servicios a través de etiquetas o incluso dándoles una puntuación.
- **Búsquedas en el repositorio.** G-Reg suministra vistas específicas para los servicios, políticas y documentos WSDL/XSD proporcionando además una ventana de búsqueda avanzada que permite localizar artefactos a partir de múltiples criterios (etiquetas, comentarios, fechas de creación-actualización, etc.). La siguiente captura muestra dicha ventana.

Figura 10. Ventana de búsqueda avanzada

Fuente: Autor.

Gestión del Ciclo de Vida de los Servicios

- **Gestionar el ciclo básico de servicios.** En G-Reg la gestión del ciclo de vida es similar a Galaxy siendo también las fases por defecto: *Initialize*, *Designed*, *Created*, *Tested*, *Deployed*, *Deprecated*.
- **Integrar un *workflow* que permita la promoción de los servicios.** G-Reg permite definir, para cada fase del *workflow*, las listas de chequeo de forma que sólo se puedan promocionar los servicios cuando se cumplen todas las condiciones. Dichas listas de chequeo son fácilmente configurables. En cuanto a

la gestión de usuarios, implementa mecanismos basados en roles y permisos para el acceso a la distinta funcionalidad de la herramienta. No obstante, el nivel de granularidad no es el adecuado, por ejemplo, no se pueden tener perfiles que únicamente promocionen los servicios de una fase a otra. Igualmente se han detectado algunos errores en la gestión de permisos.

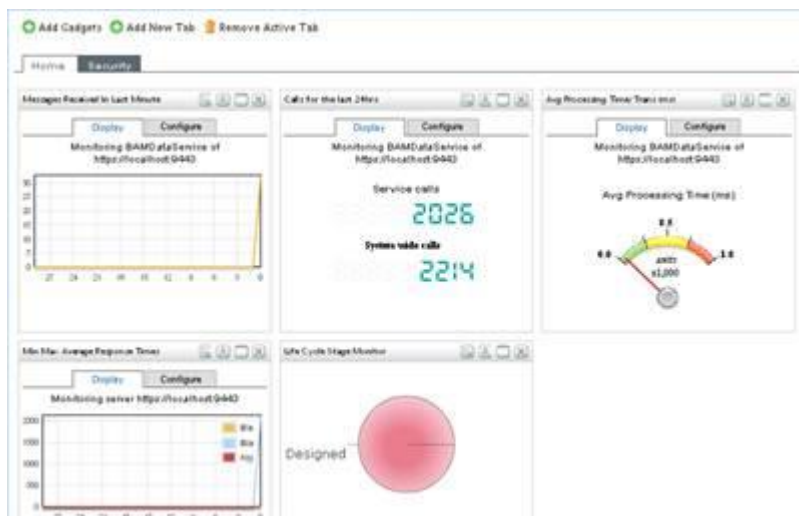
- **Administrar versiones de los servicios.** G-Reg al igual que Galaxy también permite la gestión de versiones de servicios asignando automáticamente una nueva versión cada vez que se realiza una modificación. Utiliza un mecanismo de auto-incremento para el identificador de las versiones que es común para todos los artefactos, algo que sin duda no ayuda a su identificación y trazabilidad.
- **Ejecución de pruebas de servicios.** G-Reg al igual que Galaxy tampoco tiene soporte para la ejecución de pruebas de servicios, aplicando los mismos comentarios que para Galaxy, aclarando que en este caso, se podría integrar con WSO2 Web Services Application Server - WSAS.

Políticas, Métricas, Monitorización y Configuración

- **Implementación de diversas políticas estándares.** En cuanto a G-Reg, durante la importación de WSDL y esquemas, a pesar de mostrar un mensaje indicando al usuario que se están realizando validaciones en el WSDL no avisa de errores en los mismos.
- **Adición y configuración de nuevas políticas.** G-Reg a diferencia de Galaxy, sí permite añadir nuevas políticas desde la propia consola, lo que hace más fácil su uso. Sin embargo, no parece que las tenga en cuenta a la hora de añadir nuevos servicios al registro.

- **Métricas para monitorizar los servicios en ejecución.** G-Reg proporciona métricas interesantes relacionadas con la ejecución de los servicios como “Mensajes recibidos en el último minuto”, “Número de llamadas en las últimas 24 horas”, “Tiempos medios, máximo y mínimo de ejecución” o incluso métricas de seguridad (autenticaciones erróneas) así como un gráfico donde aparece el estado, según el ciclo de vida, de los servicios registrados.
- **Interfaz gráfica amigable para las métricas de monitorización.** G-Reg proporciona un cuadro de mando que permite monitorizar gráficamente las métricas identificadas anteriormente. La consola es atractiva al estar implementada con *portlets* utilizando la especificación “Google Gadget”. La siguiente captura muestra dicho cuadro de mando. Desafortunadamente, dicha consola únicamente es capaz de monitorizar los servicios ejecutados dentro del propio contenedor de WS02 (WSO2 Web Services Application Server - WSAS). Lógicamente, es el propio contenedor el que mediante notificaciones alimenta los indicadores que se muestran en la consola.

Figura 11. Consola de G-Reg (Cuadro de mando)



Fuente: Autor.

Una mejor opción sería que esta herramienta actuara como un *proxy* ubicado entre los clientes y los proveedores de servicios de manera que, sin necesidad de modificar los servicios, por tanto de forma no intrusiva, fuera capaz de detectar y medir la actividad de los mismos. Esta opción daría mayor flexibilidad no obligando a utilizar un servidor concreto ni obligando a la modificación de los servicios para realizar las notificaciones oportunas a la herramienta BAM.

- **Notificar la ocurrencia de distintos eventos.** G-Reg incluye una herramienta para la gestión de notificaciones sobre eventos producidos tanto a nivel de artefacto o de colección. Los mecanismos de notificación incluyen el envío por email (a un usuario o a un grupo), así como la generación y envío texto plano/HTML o mensajes SOAP. Entre los tipos de eventos, además de la creación/modificación/borrado de elementos se incluye el cambio de estado dentro del ciclo de vida. También es posible sindicarse a través de RSS a los recursos registrados en la herramienta.

ANEXO E. RESULTADOS DE EVALUACIÓN DE PLATAFORMAS OPEN SOURCE PARA EL GOBIERNO SOA (GRAFICA)

Teniendo en cuenta todos los puntos tratados anteriormente para describir el comportamiento de cada herramienta sobre el punto de comparación específico se presenta una tabla comparativa entre las dos herramientas que se analizaron previamente.

Tabla 3. Galaxy versus G - REG

CARACTERÍSTICA	MULE GALAXY	WS02 G-REG
Registro/Repositorio		
registro de artefactos	↑	↑
API para el acceso	↑	↑
añadir información adicional (metadatos)	↑	↑
gestión de dependencias entre servicios	↑	↑
estructurar y organizar repositorio	↑	→
búsqueda ágil	→	↑
Ciclo de Vida de Servicios		
gestión del ciclo básico de vida	↑	↑
gestión de versiones	→	↓
<i>workflow</i> de promoción de los servicio	→	↑
herramientas para las ejecución de pruebas	↓	↓
Administración de Políticas, Métricas, Monitorización y Configuración		
implementación de políticas estándares	↑	↓
añadir o configurar nuevas políticas	→	↓
métricas, monitorizar la ejecución	↓	→
interfaz gráfica monitorización	↓	↑
ocurrencia de eventos en ejecución	→	↑

Fuente: Autor.

ANEXO F. PREDIC8 - MEMBRANE

Predic8, un proveedor de herramientas SOA de código abierto y servicios profesionales lanzo el servicio-repository.com, un registro en línea para servicios web. Después de una fase de prueba inicial del sitio web ya está disponible para la comunidad. No es el primer directorio de servicios web de forma gratuita, pero tiene algunas nuevas características que lo distinguen de las demás.

Membrane es la tercera herramienta a evaluar en este trabajo de campo, y después de investigar acerca de la herramienta, descargarla y ponerla en marcha con un servidor Apache Tomcat, se ha identificado que la herramienta permite:

- Registrar servicios proporcionando el WSDL alojado.
- Permite definir dependencias entre los servicios y comprueba la existencia de WS-I para el cumplimiento del servicio.
- Comprueba periódicamente la disponibilidad del servicio y provoca eventos en los cambios de disponibilidad.
- Comprueba si hay cambios en los WSDL o XSD provocando eventos cuando se detectan cambios.
- Mantiene las estadísticas sobre la disponibilidad de servicios, almacena una copia de los metadatos del servicio como WSDL, XSD.
- Es un cliente SOAP que permite a los XML entrar en un WS.
- Ayuda a la clasificación y el etiquetado de los servicios web.
- **Proceso de registro rápido.** Sólo la dirección URL del documento WSDL tiene que ingresarse para registrar un nuevo servicio. El registro analiza el WSDL y las definiciones de esquema XML a los que hacen referencia los metadatos.
- **SOAP Cliente Online.** El uso de los clientes dinámicos SOAP puede poner a prueba y evaluar los Servicios Web antes de que ellos empiecen a funcionar para un cliente. El registro genera un formulario para introducir los datos de solicitud.
- **Monitoreo de disponibilidad.** Hay un montón de sitios Web con servicios web, pero nunca se sabe si el servicio está todavía allí y continua trabajando.

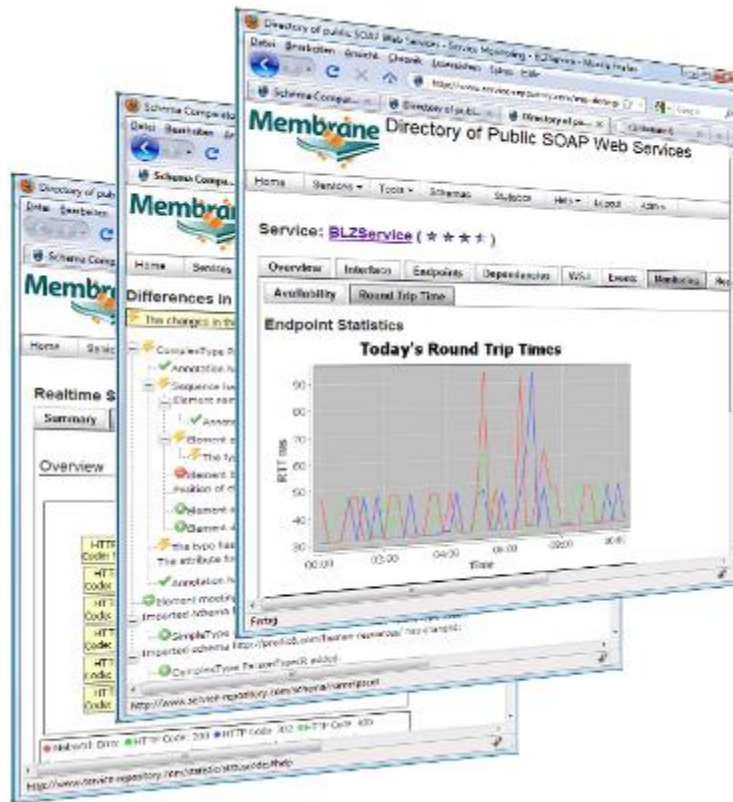
Membrane supervisa la disponibilidad de cada servicio con una regularidad definida.

- **Monitoreo del contrato WSDL.** Una causa común de fracaso en el monitoreo del contrato es el cambio de la interfaz del servicio web. Membrane observa los WSDL, las descripciones del XML de un servicio y los informes de cualquier cambio. Los consumidores de servicios web públicos conocen de inmediato si una interfaz de servicio ha cambiado.
- **Servicio “Historial Tiempo de vida”.** Durante el tiempo de vida de un servicio muchas cosas pueden suceder. El servicio puede dejar de funcionar o su interfaz puede cambiar. Todos los eventos son registrados por Membrane. El historial revela lo que sucedió y cuándo sucedió.

Algunas de las características que resultan más útiles al momento de usar la herramienta son, por ejemplo, que permite la comparación entre WSDL y esquemas XML, que chequea automáticamente cambios en WSDL y XSD y que permite etiquetar servicios y genera noticias RSS para los eventos (servicios nuevos/modificados/disponibles/no disponibles).

Por otra parte algunas de las desventajas es que carece de soporte UDDI, no tiene la capacidad de buscar servicios web basados por su nombre/descripción y no tiene la capacidad de categorizar servicios.

Figura 12. Interfaz gráfica Membrane



Fuente: MULE GALAXY¹²⁹.

Hay una versión online disponible de Membrane alojada públicamente de su herramienta de gobierno y registro en <http://www.service-repository.com/>

ANEXO G. CRITERIOS COMERCIALES

La existencia de una edición para la comunidad open source y una edición empresarial es un criterio a tener muy en cuenta a la hora de decidirse por una herramienta. Galaxy dispone de una edición Enterprise, que soporta algunas características que se echan de menos en la edición *Community*. Algunas características únicamente recogidas en la versión Enterprise son *workspaces* remotos (repositorios asociados), el balanceo de carga, la alta disponibilidad y

¹²⁹ MULE GALAXY, op. cit.

como no, una mayor y mejor documentación del producto. En la siguiente tabla se evidencian algunas características comerciales que caracterizan las herramientas trabajadas en este trabajo investigativo y nos permite enunciar algunas características comerciales de relevancia al momento de escoger una herramienta a trabajar:

Tabla 4. Galaxy versus G – REG – Evaluación de criterios comerciales

CARACTERÍSTICA	MULE GALAXY	WS02 G-REG
Tipo de Licencia	GPL2. Para algunos módulos la licencia es CPAL	Apache License 2.0
Coste de Versión <i>Community</i>	Gratuita	Gratuita
Coste de Versión <i>Enterprise</i>	7.000\$/año [9]	No se dispone de versión Enterprise
Calidad de la Documentación	Baja (<i>Community</i>)	Buena
Soporte Multi-idioma	No	No
Acceso al Código Fuente	Si	Si
Comunidad activa	Si (publicando nuevas versiones cada 3 meses)	Si (publicando nuevas versiones cada 5 meses)
Proveedores para el soporte	Limitada (la propia compañía Mule)	Limitada (la propia compañía WS02)

Fuente: Autor.